

DEFINICIÓN DE UNA APLICACIÓN MULTIPLATAFORMA EN DISPOSITIVOS MÓVILES PARA EVENTOS ACCESIBLES

Trabajo de Fin de Grado

Ingeniería Informática

Universidad Carlos III de Madrid

Autor: Paula Laplana Martín

Tutor: Israel González Carrasco

Fecha de entrega: 21 de febrero de 2019

Fecha de defensa: 11 de marzo de 2019

Tabla de contenido

1. INTRODUCCIÓN	12
1.1 Contexto social actual	12
1.2 Motivación	13
1.3 Objetivos del proyecto.....	13
1.4 Estructura del documento.....	14
2. DEFINICIONES Y ACRÓNIMOS	16
2.1 Definiciones.....	16
2.2 Acrónimos	16
3. GESTIÓN DEL PROYECTO.....	17
3.1 Ciclo de vida	17
3.1.1 Planificación de sistemas de información	17
3.1.2 Desarrollo de sistemas de información.....	17
3.1.3 Mantenimiento de sistemas de información	18
3.2 Organización.....	18
3.2.1 Estimación de recursos iniciales.....	19
Recursos Hardware.....	19
Recursos Software	19
Personal	19
3.2.2 Organización del trabajo	19
3.3 Planificación	20
3.3.1 Diagrama de Gantt	20
3.3.2 Gestión de desviaciones.....	21
4. ESTUDIO DE VIABILIDAD DEL SISTEMA.....	22
4.1 Establecimiento del alcance del sistema	22
4.1.1 Estudio de la solicitud	22
4.1.2 Identificación del alcance del sistema.....	22
4.1.2.1 Alcance del sistema: usuario registrado	23
4.1.2.1 Alcance del sistema: usuario invitado	25
4.1.3 Identificación de los interesados del sistema	27
4.2 Marco Regulador	27
4.3 Estudio de la situación actual.....	28
4.3.1 Valoración de la situación actual	28
4.3.1.1 Aplicaciones competidoras.....	28
Agenda Cultural Accesible -- ACAccesible	28
TOUR4all – PREDIF	28
4.3.1.2 Problema de la situación actual.....	29
4.4 Estudio de las alternativas de diseño	29
4.4.1 Primera alternativa: Apache Córdoba	30
4.4.2 Segunda alternativa: Xamarin	30
4.5 Valoración de las alternativas de diseño	31
4.5.1 Apache Córdoba.....	31

4.5.2 Xamarin.....	32
4.6 Selección de la alternativa de diseño	33
4.7 Entorno socio-económico	33
4.7.1 Estimación de costes.....	34
4.7.1.1 Componentes Software	34
4.7.1.2 Equipo Hardware	34
4.7.2 Estimación de personal	35
4.7.3 Estimación de presupuesto	36
4.7.3.1 Sueldo del personal	36
4.7.3.2 Equipo informático	37
4.7.3.3 Viajes y dietas	37
4.7.3.4 Costes indirectos.....	38
4.7.3.5 Margen de riesgo y beneficio	39
4.7.3.6 Estimación total del presupuesto del proyecto.....	39
5. ANALISIS DEL SISTEMA	40
5.1 Definición del sistema.....	40
5.1.1 Especificación de normas y estándares	40
5.1.2 Especificación del entorno operacional	40
5.2 Especificación de casos de uso	41
5.2.1 Diagramas de casos de uso	41
5.2.1.1 Escenario 1: Acceso a la aplicación.....	41
5.2.1.2 Escenario 2: Navegación por la aplicación.....	42
5.2.1.3 Escenario 3: Filtro de eventos.....	43
5.2.1.4 Escenario 4: Almacenamiento de datos	44
5.2.2 Definición de casos de uso	44
5.2.2.1 Casos de uso de acceso a la aplicación	45
5.2.2.2 Casos de uso de navegación por la aplicación	47
5.2.3.3 Casos de uso de filtro de eventos.....	54
5.2.3.4 Casos de uso de almacenamiento de datos.....	57
5.3 Establecimiento de requisitos software	58
5.3.1 Identificación de requisitos funcionales	59
5.3.2 Identificación de requisitos no funcionales	75
5.3.3 Identificación de requisitos inversos.....	80
5.3.4 Identificación de requisitos de interfaz.....	82
5.4 Identificación de subsistemas de análisis	86
5.4.1 Definición de cada subsistema.....	87
5.4.1.1 Acceso a la aplicación	87
5.4.1.2 Navegación por la aplicación	87
5.4.1.3 Filtro de eventos.....	87
5.4.1.4 Almacenamiento de datos.....	87
5.4.2 Relaciones entre subsistemas	88
5.5 Elaboración del modelo de datos	88
5.6 Definición de las interfaces de usuario.....	89
5.6.1 Interfaces de acceso a la aplicación	90
5.6.2 Interfaces de navegación por la aplicación	94
5.6.3 Interfaces de filtro de eventos	99
5.4.3.1 Interfaces de modo invitado.....	103
5.6.4 Interfaces de almacenamiento de datos.....	105

5.7 Consistencia de las interfaces de usuario	106
5.7.1 Acceso a la aplicación: Casos de uso - interfaces de usuario	107
5.7.2 Navegación por la aplicación: Casos de uso – interfaces de usuario	107
5.7.3 Filtro de eventos: Casos de uso - interfaces de usuario.....	108
5.7.4 Almacenamiento de datos: Casos de uso - interfaces de usuario.....	108
6. DISEÑO DEL SISTEMA.....	109
6.1 Objetivo.....	109
6.2 Arquitectura del sistema.....	109
6.2.1 Controlador	109
6.2.2 Vista	110
6.2.3 Modelo	110
6.3 Interacción de prototipos y componentes.....	112
6.3.1 Interacción prototipos y casos de uso.....	112
6.3.2 Interacción componentes e interfaces de usuario.....	113
6.4 Diseño de prototipos	116
6.4.1 Definición de los prototipos	116
Cabecera.....	117
Firebase	118
Consumir servicios Web	120
Filtro eventos	121
Funciones	122
Información evento.....	123
Leer eventos.....	124
Menú horizontal.....	125
6.5 Diseño de componentes	125
6.5.1 Definición de los componentes.....	125
6.6 Relación entre capas de arquitectura	133
6.6.1 Relación Controlador y Vista	133
6.6.2 Relación Controlador y Modelo	135
6.6.3 Relación Vista y Modelo.....	135
6.6.4 Relación Vista.....	136
6.7 Diseño físico de datos	136
7. VALIDACIÓN DEL SISTEMA.....	139
7.1 Definición del plan de pruebas.....	139
7.1.1 Pruebas de acceso a la aplicación	140
7.1.2 Pruebas de navegación por la aplicación	147
7.1.3 Pruebas de filtro de eventos	155
7.1.4 Pruebas de almacenamiento de datos.....	158
8. IMPLANTACIÓN DEL SISTEMA.....	162
9. CONCLUSIONES.....	163
9.1 Posibles mejoras del sistema	164
10. BIBLIOGRAFÍA.....	165
ANEXO I: MANUAL DE USO.....	167

1. Manual de usuario.....	167
1.1 Acceso a la aplicación.....	167
1.2 Registrarse en la aplicación.....	168
1.3 Iniciar sesión en la aplicación.....	168
1.4 Pantalla principal de la aplicación.....	169
1.5 Selección y filtro de eventos.....	170
1.6 Marcar favorito un evento.....	173
1.7 Desmarcar evento como favorito.....	174
1.7 Volver a pantalla principal.....	174
1.8 Acceder y modificar el perfil del usuario.....	174
1.8.1 Acceder al perfil.....	174
1.8.2 Modificar perfil.....	175
1.9 Información sobre CESyA.....	176
1.10 Cerrar sesión.....	177
1.11 Entrar en la aplicación sin registrarse.....	177
1.11.1 Acceder a pantalla principal sin haberse registrado.....	177
1.11.2 Acceder a información de la aplicación.....	178
1.11.3 Seleccionar eventos sin haberse registrado.....	179
1.11.4 Volver a pantalla principal.....	179
ABSTRACT	181
1. Introduction	181
1.1 Current social context.....	181
1.2 Motivation.....	182
1.3 Project Objectives.....	182
1.4 Document Structure.....	183
2. Feasibility study of the system.....	184
2.1 Setting the scope of the system.....	184
2.1.1 Study of the application.....	184
2.1.2 Identification of the scope of the system.....	184
2.1.2.1 scope of system: Registered user.....	184
2.1.2.2 Scope of System: Guest user.....	187
2.2 Study of the current situation.....	189
2.2.1 Assessing the current situation.....	189
2.2.1.1 Competing applications.....	189
Accessible Cultural Agenda-- Accessible.....	189
TOUR4all – PREDIF.....	189
2.2.1.2 problem of the current situation.....	189
3. System Analysis.....	190
3.1 System Definition.....	190
3.2 Identification of analysis subsystems.....	191
3.2.1 Definition of each subsystem.....	191
3.2.1.1 access to the application.....	191
3.2.1.2 Navigation by Application.....	192
3.2.1.3 Event Filter.....	192
3.2.1.4 Data storage.....	192
4. System Design	192
4.1 Objective.....	192
4.2 System architecture.....	193
5. Conclusions	193

5.1 Possible system upgrades	194
------------------------------------	-----

Índice de tablas

Tabla 1: Interesados en el sistema	27
Tabla 2: Criterios selección alternativa.....	33
Tabla 3: Estimación costes Software	34
Tabla 4: Estimación costes Hardware	35
Tabla 5: Estimación total de costes	35
Tabla 6: Estimación de personal	36
Tabla 7: Estimación €/horas por tarea	36
Tabla 8: Estimación sueldo personal	37
Tabla 9: Estimación viajes y dietas	38
Tabla 10: Margen de riesgo y beneficio.....	39
Tabla 11: Definición de casos de uso.....	44
Tabla 12: Abrir la aplicación CU-00.....	45
Tabla 13: Entrar sin registrarse CU-01.....	45
Tabla 14: Registrarse CU-02	46
Tabla 15: Iniciar sesión CU-03.....	47
Tabla 16: Entrar en “Categorías” CU-04	47
Tabla 17: Seleccionar una categoría CU-05	47
Tabla 18: Entrar en Mi Archivo CU-06	48
Tabla 19: Seleccionar un evento guardado CU-07.....	48
Tabla 20: Seleccionar un evento guardado CU-08.....	49
Tabla 21: Entrar en CESyA CU-09.....	49
Tabla 22: Entrar en “Eventos en...” CU-10.....	49
Tabla 23: Seleccionar un evento CU-11.....	50
Tabla 24: Acceder a web de un evento CU-12.....	50
Tabla 25: Volver a ventana principal CU-13	51
Tabla 26: Volver a ventana anterior CU-14	51
Tabla 27: Desplegar el menú lateral CU-16	52
Tabla 28: Acceder a Mi Perfil CU-17	52
Tabla 29: Editar perfil CU-18.....	52
Tabla 30: Cerrar sesión CU-19	53
Tabla 31: Contactar con CESyA CU-20	53
Tabla 32: Entrar en sección “Acerca de...” CU-21.....	54
Tabla 33: Salir del menú lateral CU-22	54
Tabla 34: Filtrar evento por ciudad CU-23.....	54
Tabla 35: Seleccionar una ciudad popular CU-24	55
Tabla 36: Buscar una ciudad CU-25	55
Tabla 37: Seleccionar una ciudad CU-26	56
Tabla 38: Filtrar eventos por accesibilidad CU-27	56
Tabla 39: Aplicar filtros CU-28	56
Tabla 40: Aplicar filtros CU-29	57
Tabla 41: Guardar evento como favorito CU-30.....	57
Tabla 42: Desmarcar evento como favorito CU-31	58

Tabla 43: Definición de requisitos software	58
Tabla 44: RF-00	60
Tabla 45: RF-01	60
Tabla 46: RF-02	60
Tabla 47: RF-03	61
Tabla 48: RF-04	61
Tabla 49: RF-05	62
Tabla 50: RF-06	62
Tabla 51: RF-07	63
Tabla 52: RF-08	63
Tabla 53: RF-09	63
Tabla 54: RF-10	64
Tabla 55: RF-11	64
Tabla 56: RF-12	65
Tabla 57: RF-13	65
Tabla 58: RF-14	65
Tabla 59: RF-15	66
Tabla 60: RF-16	66
Tabla 61: RF-17	67
Tabla 62: RF-18	67
Tabla 63: RF-19	67
Tabla 64: RF-20	68
Tabla 65: RF-21	68
Tabla 66: RF-22	69
Tabla 67: RF-23	69
Tabla 68: RF-24	70
Tabla 69: RF-25	70
Tabla 70: RF-26	70
Tabla 71: RF-27	71
Tabla 72: RF-28	71
Tabla 73: RF-29	72
Tabla 74: RF-30	72
Tabla 75: RF-31	72
Tabla 76: RF-32	73
Tabla 77: RF-33	73
Tabla 78: RF-34	74
Tabla 79: RF-35	74
Tabla 80: RF-36	75
Tabla 81: RF-37	75
Tabla 82: RNF-00	75
Tabla 83: RNF-01	76
Tabla 84: RNF-02	76
Tabla 85: RNF-03	77
Tabla 86: RNF-04	77
Tabla 87: RNF-05	78
Tabla 88: RNF-06	78
Tabla 89: RNF-07	78
Tabla 90: RNF-08	79

Tabla 91: RNF-09	79
Tabla 92: RNF-10	80
Tabla 93: RNF-11	80
Tabla 94: RINV-00	81
Tabla 95: RINV-01	81
Tabla 96: RINV-02	81
Tabla 97: RINV-03	82
Tabla 98: RINT-00	82
Tabla 99: RINT-01	83
Tabla 100: RINT-02	83
Tabla 101: RINT-03	84
Tabla 102: RINT-04	84
Tabla 103: RINT-05	84
Tabla 104: RINT-06	85
Tabla 105: RINT-07	85
Tabla 106: RINT-08	86
Tabla 107: RINT-09	86
Tabla 108: Consistencia CU-IU Acceso a la aplicación	107
Tabla 109: Consistencia CU-IU Navegación por la aplicación	108
Tabla 110: Consistencia CU-IU Filtro de eventos	108
Tabla 111: Consistencia CU-IU Almacenamiento de datos	108
Tabla 112: Trazabilidad IU- CT 1	114
Tabla 113: Trazabilidad IU- CT 2	115
Tabla 114: Trazabilidad IU- CT 3	116
Tabla 115: Definición de prototipos	117
Tabla 116: Prototipo Consumir servicios Web PT-02	121
Tabla 117: Prototipo Filtro eventos PT-03	122
Tabla 118: Prototipo Funciones PT-04	123
Tabla 119: Prototipo Información evento PT-05	124
Tabla 120: Prototipo Leer eventos PT-06	125
Tabla 121: Prototipo Menú horizontal PT-07	125
Tabla 122: Componente Index CT-00	126
Tabla 123: Componente Registro CT-01	127
Tabla 124: Componente Login CT-02	127
Tabla 125: Componente Inicio CT-03	127
Tabla 126: Componente Inicio Sin Registrar CT-04	127
Tabla 127: Componente Eventos Sin Registrar CT-05	128
Tabla 128: Componente Configuración CT-06	128
Tabla 129: Componente Perfil CT-07	128
Tabla 130: Componente Eventos guardados CT-08	129
Tabla 131: Componente Eventos CT-09	129
Tabla 132: Componente Información Eventos CT-10	129
Tabla 133: Componente Filtro Eventos CT-11	129
Tabla 134: Componente Estilo Cabecera CT-12	130
Tabla 135: Componente Estilo Eventos CT-13	130
Tabla 136: Componente Estilo Filtro Eventos CT-14	130
Tabla 137: Componente Estilo Index CT-15	131
Tabla 138: Componente Estilo Información Eventos CT-16	131

Tabla 139: Componente Estilo Inicio CT-17	131
Tabla 140: Componente Estilo Menú Horizontal CT-18	132
Tabla 141: Componente Estilo Perfil CT-19	132
Tabla 142: Componente Estilo Registro CT-20	132
Tabla 143: Componente Usuarios CT-21	132
Tabla 144: Componente Servicios Web CT-22.....	133
Tabla 145: Relación Controlador – Vista I.....	134
Tabla 146: Relación Controlador - Vista II	134
Tabla 147: Relación Controlador - Vista III	134
Tabla 148: Relación Controlador – Modelo	135
Tabla 149: Relación Vista – Modelo I	135
Tabla 150: Relación Vista - Modelo II	135
Tabla 151: Relación Vista - Modelo III	136
Tabla 152: Colección Users.....	138
Tabla 153: Definición plan de pruebas	139
Tabla 154: PAA-00	140
Tabla 155: PAA-01	140
Tabla 156: PAA-02	141
Tabla 157: PAA-03	141
Tabla 158: PAA-04	141
Tabla 159: PAA-05	142
Tabla 160: PAA-06	142
Tabla 161: PAA-07	143
Tabla 162: PAA-08	143
Tabla 163: PAA-09	143
Tabla 164: PAA-10	144
Tabla 165: PAA-11	144
Tabla 166: PAA-12	145
Tabla 167: PAA-13	145
Tabla 168: PAA-14	145
Tabla 169: PAA-15	146
Tabla 170: PAA-16	146
Tabla 171: PAA-17	146
Tabla 172: PAA-18	147
Tabla 173: PAA-19	147
Tabla 174: PNA-00.....	148
Tabla 175: PNA-01	148
Tabla 176: PNA-02	148
Tabla 177: PNA-03	149
Tabla 178: PNA-04	149
Tabla 179: PNA-05	149
Tabla 180: PNA-06	150
Tabla 181: PNA-07	150
Tabla 182: PNA-08	150
Tabla 183: PNA-09	151
Tabla 184: PNA-10	151
Tabla 185: PNA-11	152
Tabla 186: PNA-12	152

Tabla 187: PNA-13	152
Tabla 188: PNA-14	153
Tabla 189: PNA-15	153
Tabla 190: PNA-16	153
Tabla 191: PNA-17	154
Tabla 192: PNA-18	154
Tabla 193: PNA-19	154
Tabla 194: PNA-20	155
Tabla 195: PFE-00	155
Tabla 196: PFE-01	156
Tabla 197: PFE-02	156
Tabla 198: PFE-03	156
Tabla 199: PFE-04	157
Tabla 200: PFE-05	157
Tabla 201: PFE-06	158
Tabla 202: PFE-07	158
Tabla 203: PFE-08	158
Tabla 204: PAD-00	159
Tabla 205: PAD-01	159
Tabla 206: PAD-02	159
Tabla 207: PAD-03	160
Tabla 208: PAD-04	160
Tabla 209: PAD-05	161
Tabla 210: PAD-06	161

Tabla de ilustraciones

Ilustración 1: Dominio de Android e IOS (3)	12
Ilustración 2: Procesos Métrica V3	18
Ilustración 3: Diagrama de Gantt.....	20
Ilustración 4: Alcance del sistema usuario registrado	25
Ilustración 5: Alcance del sistema usuario invitado.....	26
Ilustración 6: Arquitectura Apache Córdoba (13)	30
Ilustración 7: Arquitectura Xamarin (15)	31
Ilustración 8: Diagrama CU Acceso a la aplicación	42
Ilustración 9: Diagrama CU Navegación por la aplicación I	42
Ilustración 10: : Diagrama CU Navegación por la aplicación II	43
Ilustración 11: Diagrama CU Filtro de eventos	43
Ilustración 12: Diagrama CU Almacenamiento de datos.....	44
Ilustración 13: Patrón MVC del sistema	87
Ilustración 14: Relación entre los subsistemas.....	88
Ilustración 15: Modelo de datos del sistema.....	89
Ilustración 16: Pantalla inicial de la aplicación IU-00.....	91
Ilustración 17: Inicio de sesión IU-01.....	92
Ilustración 18: Registro 1 IU-02	93
Ilustración 19: Registro 2 IU-03	93

Ilustración 20: Pantalla principal IU-04.....	94
Ilustración 21: Mi Archivo IU-05	95
Ilustración 22: CESyA IU-06	95
Ilustración 23: Menú lateral IU-07.....	96
Ilustración 24: Mi Perfil IU-08.....	97
Ilustración 25: Eventos IU-09.....	98
Ilustración 26: Información evento IU-10.....	99
Ilustración 27: Filtrar evento IU-11.....	100
Ilustración 28: Seleccionar ciudad popular IU-12	100
Ilustración 29: Buscar ciudad IU-13	100
Ilustración 30: Seleccionar solo filtro ciudad IU-11	101
Ilustración 31: Seleccionar ciudad y accesibilidad IU-11	101
Ilustración 32: Filtros aplicados sobre eventos 1 IU-12.....	102
Ilustración 33: Filtros aplicados sobre eventos 2 IU-12.....	102
Ilustración 34: Ciudad sin eventos disponibles IU-13	103
Ilustración 35: Inicio modo invitado IU – 14.....	103
Ilustración 36: Eventos en modo invitado IU – 16.....	103
Ilustración 37: Mi Archivo Usuario invitado IU - 15.....	104
Ilustración 38: Guardar eventos usuario invitado IU -17.....	105
Ilustración 39: Configuración II CU- 18	105
Ilustración 40: Configuración IU - 18	105
Ilustración 41: Evento favorito IU-19.....	106
Ilustración 42: Evento guardado en Mi Archivo IU-05.....	106
Ilustración 43: Capa controlador MVC	110
Ilustración 44: Capa vista MVC	110
Ilustración 45: Capa Modelo MVC.....	111
Ilustración 46: Arquitectura del sistema	111
Ilustración 47: Prototipo Cabecera PT-00.....	117
Ilustración 48: Prototipo Firebase PT-01	120
Ilustración 49: Relación entre componentes Vista	136
Ilustración 50: Diagrama E-R Usuarios y Eventos	137
Ilustración 51: Implantación en Android	162
Ilustración 52: Implantación en IOS.....	162
Ilustración 53: Manual Usuario - Pantalla inicial	167
Ilustración 54: Manual Usuario - Registro I	168
Ilustración 55: Manual Usuario – Registro II	168
Ilustración 56: Manual Usuario - Inicio de sesión.....	169
Ilustración 57: Manual de Usuario – Opciones pantalla principal	170
Ilustración 58: Manual de usuario - Eventos	170
Ilustración 59: Manual Usuario - Filtro de eventos	171
Ilustración 60: Manual Usuario - Filtro de eventos	172
Ilustración 61: Manual Usuario - Información de un evento.....	173
Ilustración 62: Manual Usuario - Marcar favorito	173
Ilustración 63: Manual Usuario - Volver a pantalla principal	174
Ilustración 64: Manual Usuario - Acceso a Mi Perfil.....	175
Ilustración 65: Manual Usuario - Editar datos del perfil.....	175
Ilustración 66: Manual Usuario - Acceso a Configuración	176
Ilustración 67: Manual Usuario - Cerrar sesión	177

Ilustración 68: Manual Usuario - pantalla principal usuario invitado.....	178
Ilustración 69: Manual Usuario - Eventos de usuario invitado.....	179
Ilustración 70: Manual Usuario - Volver a pantalla principal usuario invitad.....	180
Illustration 71: Android and IOS Domain.....	181
Illustration 72: Scope of the registered user system.....	187
Illustration 73: User System Scope Guest.....	188
Illustration 74: System MVC Pattern	191
Illustration 75: Relationship between the subsystems.....	192

1. INTRODUCCIÓN

La finalidad de este documento es presentar el desarrollo de un trabajo de fin de estudios del grado Ingeniería Informática. Se trata de un trabajo académico donde se ponen en práctica muchos de los conocimientos aprendidos durante la carrera.

El trabajo realizado consta de la definición de una aplicación multiplataforma en dispositivos móviles para eventos accesibles.

1.1 Contexto social actual

¿Por qué una aplicación multiplataforma? Esta es una de las preguntas que surgen cuando se inicia un desarrollo de aplicación móvil.

Hoy en día, el mercado de marcas móviles está muy dividido, y no por el diseño estético que pueda ofrecer una marca concreta, sino por el sistema operativo que incluya el móvil. Se cuentan con más de 8 diferentes sistemas operativos para dispositivos móviles, como por ejemplo Android, IOS, Windows Phone, BlackBerry, Symbian, etc (1). Bien es cierto que en la actualidad los más predominantes son *Android*, propio de Google e *IOS*, de Apple.

La batalla entre Android e IOS es algo que lleva mucho tiempo en marcha, y que no acabará (2). Asíque, siguiendo la información que aporta el párrafo anterior, se podrían descartar el resto de sistemas operativos como competencia en el mercado, quedando estos dos líderes.

Analizando la sociedad, hay mucha gente que prefiere Android sobre IOS, y otro porcentaje de gente que prefiere IOS sobre Android. Actualmente, este es el porcentaje de usuarios que usan cada plataforma:

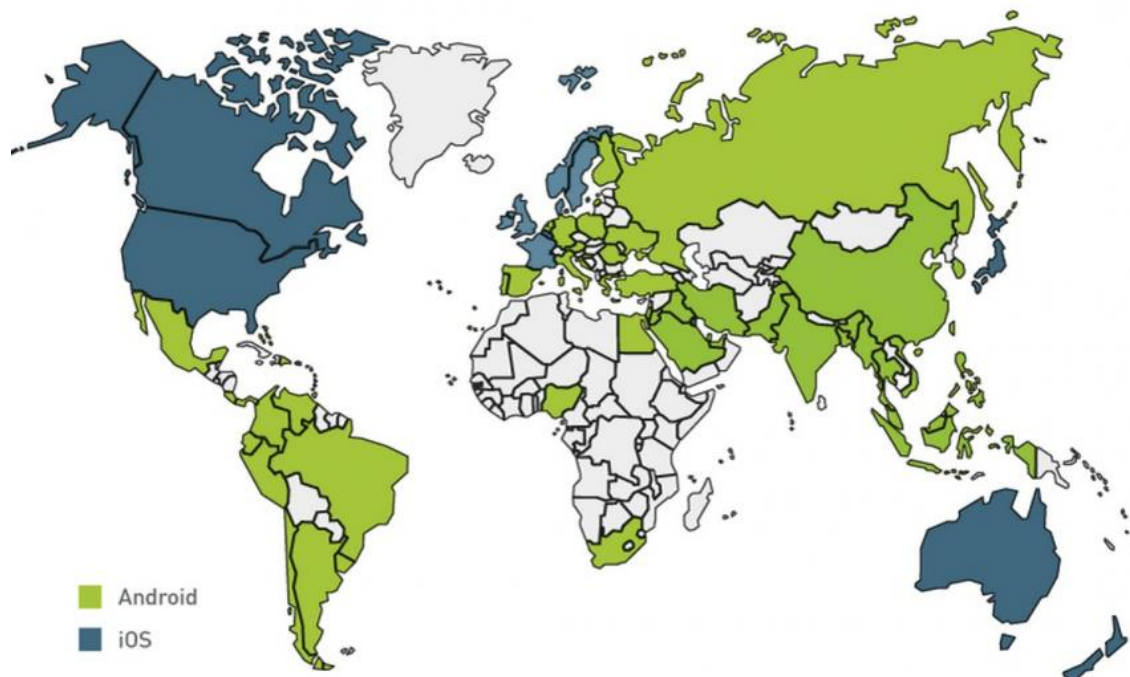


Ilustración 1: Dominio de Android e IOS (3)

Con la ilustración 1 queda claro que Android es bastante más usado por los usuarios que iOS. Esto tiene una explicación clara y es que, alejado de las preferencias de cada usuario, Android es un sistema operativo que esta implantado en la mayoría de marcas de móviles del mercado. Es por ello que, por probabilidad, hay más móviles con este sistema.

Esta lucha incesante entre dichos sistemas operativos lleva a plantearse para qué plataforma desarrollar una aplicación móvil. Todo buen desarrollador podría crear un proyecto nativo para cada sistema operativo, y así ampliar el abanico de posibilidades de venta en el mercado. No obstante, ¿por qué no crear una aplicación donde un solo proyecto software sirva para todas las plataformas que se desee? Es por ello que existe el desarrollo multiplataforma, que, además de cubrir este problema minoritario, aporta muchas ventajas al desarrollo móvil.

Conociendo este análisis de la situación actual del desarrollo de aplicaciones, la idea de definir una aplicación para eventos accesibles va de la mano de que ésta sea multiplataforma. Además, las plataformas en las que sería oportuno desarrollar la aplicación son las líderes del mercado, Android e iOS.

1.2 Motivación

La principal motivación para crear una aplicación de eventos accesibles ha sido el tutor de este proyecto. En la actualidad, existen aplicaciones que proporcionan eventos con servicios de accesibilidad, pero no cuentan con un desarrollo nativo.

Es por ello por lo que, conociendo el contexto de la definición de aplicaciones móviles, se ha querido apostar por la creación de un desarrollo híbrido. Las principales ventajas que va a cubrir son la mejora en el mantenimiento, rendimiento y escalabilidad de un proyecto.

Otra de las ventajas por la que se ha escogido este modelo de desarrollo, es porque está orientado a la distribución de contenido (4) y la aplicación que se quiere definir distribuye el contenido de qué eventos disponibles hay en diferentes lugares.

Además, otro motivo para realizar este proyecto es el aprendizaje y la familiarización con este nuevo paradigma de creación de aplicaciones, ya que está en pleno crecimiento.

Este proyecto va a abarcar usuarios tanto de Android como de iOS, sin tener que correr el riesgo de no poder salir al mercado en alguna de estas plataformas. Si este desarrollo hubiese sido nativo, aún pudiendo crear varios sistemas nativos de la misma aplicación, tal vez se habría escogido una única plataforma, por la simplicidad y sencillez en cuando a desarrollo y mantenimiento del código.

1.3 Objetivos del proyecto

Los objetivos a cubrir con este proyecto son los siguientes:

- Aprender a desarrollar aplicaciones para dispositivos móviles
- Aprender el paradigma de creación de aplicaciones multiplataforma
- Abarcar los dos sistemas operativos móviles líderes del mercado
- Aplicar los máximos conocimientos de un desarrollo software aprendidos en el grado

- Crear una aplicación que, proporcionando eventos accesibles, pueda desarrollarse y mantenerse de una manera óptima
- Crear una aplicación fácil y simple de usar para usuarios con necesidades de accesibilidad
- Mostrar los eventos disponibles que hay en una determinada ciudad con un tipo de accesibilidad concreta
- Permitir acceder a la aplicación registrándose y sin registrar, pero aplicando restricciones sobre los usuarios que no se registren
- Almacenar información personal de los usuarios que utilicen la aplicación, bajo protección de los datos
- Obtener información de los eventos mediante peticiones a servicios web

1.4 Estructura del documento

Este documento se compone de las siguientes partes:

- **Introducción:** proporciona una visión global del proyecto, mostrando las principales ideas que han movido este desarrollo y analizando el contexto actual. También se estudian los objetivos que ha de conseguir este proyecto. Incluye la descripción de las partes que componen este documento.
- **Definición y acrónimos:** este apartado incluye la explicación de términos que pueden ser técnicos y desconocidos por el lector y los acrónimos y/o abreviaturas que se han usado durante todo el documento.
- **Gestión del proyecto:** se especifica de forma detalla cada tarea a realizar en este proyecto, contando el ciclo de vida, la organización y la planificación del mismo, para lograr un producto software de calidad.
- **Estudio de viabilidad del sistema:** en este apartado se estudia el contexto en el que se encuentra el desarrollo de este sistema, comparándolo con aplicaciones competidoras y analizando posibles alternativas de desarrollo de la misma. Se escogerá, en este apartado, la opción para llevar a cabo el proyecto.
- **Análisis del sistema:** se realizará un análisis previo al desarrollo de la aplicación, estudiando su definición, las necesidades del usuario para crear este proyecto, se elabora los modelos de datos que se van a seguir y se diseñarán las interfaces de usuario de la aplicación. Todo esto se valida con matrices de trazabilidad para comprobar la consistencia del sistema.
- **Diseño del sistema:** en este apartado se construye lo que anteriormente se analizó para la creación de este proyecto. Se define la arquitectura del sistema, la interacción entre cada elemento de su estructura y se diseña el modelo físico que seguirá la base de datos.
- **Validación del sistema:** se creará un plan de pruebas para validar que el producto software actúa como se espera.
- **Implantación del sistema:** en este apartado se detalla la implantación del sistema, una vez desarrollado y validado.
- **Conclusiones:** este apartado expone las conclusiones que el alumno ha sacado del desarrollo de este proyecto.
- **Bibliografía:** en este apartado se listan todas las páginas web o librerías que el alumno ha empleado y en las que se ha apoyado para el desarrollo de este proyecto software.

- **Manual de uso:** este documento contiene un manual de usuario para que sepan y puedan utilizarla, indicando paso a paso las acciones que permite la aplicación, evitando que se encuentre en situaciones de error.
- **Abstract:** parte escrita en inglés donde se narra la introducción, resumen y conclusiones del proyecto.

2. DEFINICIONES Y ACRÓNIMOS

Este apartado consta de las definiciones y acrónimos que componen el documento.

2.1 Definiciones

- **Desarrollo multiplataforma/híbrido:** desarrollo de un proyecto que, con un solo código, sirva para varias plataformas
- **Metodología Métrica V3:** metodología de planificación, desarrollo y mantenimiento de sistemas de información, promovida por el ministerio de Hacienda y Función Pública del Gobierno de España (5).
- **Diagrama de Gantt:** herramienta para planificar y programar tareas a lo largo de un periodo determinado (6).
- **Base de datos realtime:** base de datos que gestiona la carga, la actualización, la consulta y el borrado de datos en tiempo real para cada cliente.
- **CESyA:** Centro Español del Subtitulado y la Audiodescripción. Centro dependiente del Real Patronato sobre Discapacidad del Ministerio de Sanidad, Servicios Sociales e igualdad, gestionada por la Universidad Carlos III de Madrid (7).
- **Entorno opensource:** modelo de desarrollo de software basado en la codificación abierta.
- **Firebase:** plataforma de desarrollo en la nube de Google.
- **Realtime Database de Firebase:** base de datos en tiempo real que Firebase proporciona
- **Stakeholder:** interesados y/o involucrados en el desarrollo de un producto software
- **Play Store:** tienda de aplicaciones móviles para dispositivos con sistema operativo Android
- **App Store:** tienda de aplicaciones móviles para dispositivos con sistema operativo IOS

2.2 Acrónimos

- **EVS:** Estudio de Viabilidad del Sistema
- **ASI:** Análisis del Sistema de Información
- **DSI:** Diseño del Sistema de Información
- **CSI:** Construcción del Sistema de Información
- **IAS:** Implantación y aceptación del Sistema
- **CESyA:** Centro Español del Subtitulado y la Audiodescripción
- **ACA:** Agenda Cultural Accesible
- **TFG:** Trabajo de Fin de Grado
- **UC3M:** Universidad Carlos III de Madrid
- **MVC:** Modelo Vista Controlador

3. GESTIÓN DEL PROYECTO

3.1 Ciclo de vida

(8) Este proyecto se va a llevar a cabo a partir de una adaptación de la metodología Métrica V3.

Esta metodología está orientada a la planificación, desarrollo y mantenimiento de sistemas de información. Plantea este marco de gestión para asegurar que los proyectos cumplen sus objetivos en términos de calidad, costes y plazos.

Siguiendo esta metodología, por tanto, se consiguen estos hitos:

- Proporcionar y definir un sistema de información que ayude a conseguir los fines de la organización creadora.
- Dotar a la organización del producto software esperado que satisfaga las necesidades de los usuarios, dando una gran importancia al análisis de los requisitos.
- Mejorar la productividad del equipo, permitiendo gran capacidad de adaptación a los cambios.
- Facilitar el uso y mantenimiento del producto software obtenido.

Se trata de una metodología orientada al proceso, adecuándose de mejor forma a la entrada-transformación-salida que se produce en cada uno de los componentes que forman el ciclo de vida de un proyecto.

Esta metodología, con una única estructura, cubre distintos tipos de desarrollo, facilitando mediante las interfaces la realización de los procesos de apoyo y organizativos: gestión de proyectos, gestión de configuración y aseguramiento de calidad y seguridad.

La metodología Métrica V3 se divide en tres grandes procesos:

3.1.1 Planificación de sistemas de información

Este proceso pretende adaptar el proyecto a la competitividad y cambio al que están sometidas las organizaciones, siguiendo nuevas exigencias con la velocidad que demande el entorno.

La existencia de nuevas tecnologías permite a los sistemas soportar la toma de decisiones a grandes volúmenes de información.

3.1.2 Desarrollo de sistemas de información

Este proceso es altamente complejo, y por ello esta metodología lo ha dividido en varios bloques para facilitar su comprensión:

- **Estudio de Viabilidad del Sistema (EVS):** se estudia cómo de viable es la propuesta a desarrollar, analizando la situación actual del contexto del producto.
- **Análisis del Sistema de Información (ASI):** se analizan las necesidades del usuario que el proyecto pretende cubrir, estudiando las herramientas y componentes a seguir.

- **Diseño del Sistema de Información (DSI):** se diseña el sistema que anteriormente se ha analizado, definiendo la arquitectura del sistema.
- **Construcción del Sistema de Información (CSI):** desarrollo y creación del sistema diseñado.
- **Implantación y Aceptación del Sistema (IAS):** puesta en producción del sistema y aceptación por parte del cliente.

3.1.3 Mantenimiento de sistemas de información

Finalmente, este proceso comprende las tareas de modificación de los componentes del sistema.

A continuación, se va a mostrar una imagen de los procesos que sigue esta metodología:



Ilustración 2: Procesos Métrica V3

Este proyecto va a seguir una adaptación de esta metodología, por tratarse de un proyecto académico. El ciclo de vida por tanto que va a seguir es el siguiente:

3.2 Organización

Para la realización de este proyecto se ha de organizar las tareas que lo componen. También es necesario hacer una estimación inicial de los recursos, equipo y actividades necesarias que hay que desarrollar.

Este proyecto es un trabajo académico de fin de grado, lo que implica que se va a realizar de manera individual, por un único alumno. Esto quiere decir que todo el proyecto se va a desempeñar por una persona, utilizando diferentes roles.

3.2.1 Estimación de recursos iniciales

Este apartado estima los recursos iniciales necesarios para el proyecto.

Recursos Hardware

Para desarrollar este proyecto se necesitará de un ordenador. También se requiere de un dispositivo móvil de diferentes sistemas operativos para poder implantar del sistema.

Recursos Software

Este proyecto necesita utilizar bastantes componentes software para su desarrollo. Requerirá de una base de datos, de programas que modelen en UML, de un ejecutable base del proyecto, de los compiladores de cada plataforma, etc. Todos estos programas software son de carácter gratuito por lo que no supondrán ningún coste económico.

Personal

Al tratarse de un TFG, el equipo que va a planificar, desarrollar y mantener este sistema es el alumno. Éste realizará las funciones de jefe de proyecto, analista, diseñador, y programador.

3.2.2 Organización del trabajo

Este apartado va a analizar la forma de organizar el trabajo para hacer un sistema de calidad y llevar a cabo la toma de decisiones de una forma estructurada.

El papel de cliente de este proyecto será desempeñado por el tutor y el de jefe de proyecto por el alumno. Tanto el cliente como el jefe de proyecto se reunirán de forma periódica y regulada para hablar del sistema software que se quiere desarrollar.

El papel de analista, que también será desempeñado por el alumno, es el encargado de extraer las necesidades del usuario, para gestionar la elicitación los requisitos del sistema. En este caso, no se tienen requisitos de usuario porque de se va a partir directamente de los requisitos de software del sistema. Además, tendrá que analizar cómo se estructura el sistema y qué componentes y paquetes lo forman.

El diseñador del proyecto, que este papel lo desempeñará también el alumno, es el encargado de construir el sistema. Tendrá que llevar a cabo el análisis previo del proyecto, construyendo así la arquitectura del sistema, y todos los elementos que la forman. Junto con el programador, también el alumno, desarrollarán e implementarán el producto final.

Una vez se tenga el producto ya desarrollado, se establecerán pruebas para comprobar la validez y calidad del sistema.

Finalmente, se pondrá en producción todo el producto y será mantenido por el administrador del sistema. Este administrador, inicialmente es el alumno, pero en cuanto el producto final esté ya implantado, será administrado por otra organización.

3.3 Planificación

Para crear una planificación real del proyecto, se va a diseñar un diagrama de Gantt donde se exponga cada tarea a desarrollar mostrando el tiempo que va a llevar completarla, desde la fecha de inicio hasta la fecha de fin.

3.3.1 Diagrama de Gantt

A continuación, se muestra el Diagrama de Gantt para estimar la planificación del este proyecto:

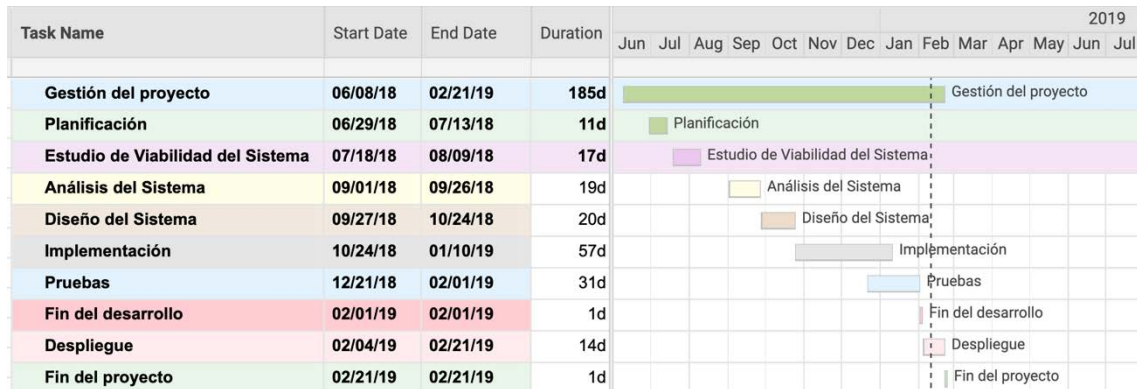


Ilustración 3: Diagrama de Gantt

Como puede observarse en el diagrama, este proyecto tendría una duración aproximada de 6 meses.

La gestión del proyecto duraría desde el principio del desarrollo hasta el final, ya que en todo momento es necesario que se controle la realización de este.

En la primera reunión que se tenga con el cliente, se va a decidir qué sistema desarrollar, cuál será el producto final y con qué herramientas se va a construir. Esto se da en la gestión del proyecto. Como en todo desarrollo software, cada actividad que se realiza puede necesitar constantes revisiones y quizá impliquen modificaciones. Es por ello por lo que la gestión del proyecto dura de principio a fin.

Durante los siguientes meses se va a estudiar con más detalle, la situación actual del contexto en el que se va a desarrollar, el entorno tecnológico, la definición de los requisitos que van a definir al proyecto y la construcción de la arquitectura entre otros. Todas estas tareas requieren de varios meses para que quede bien asentado la base de todo el desarrollo.

Una vez se ha estudiado la viabilidad, y analizado y diseñado el sistema, se lleva a cabo la implementación del mismo. Está una parte fundamental del proyecto, ya que es la realización del producto final. Por eso requiere de casi dos meses de trabajo. Dado a que no se acaba en la implementación, este periodo continúa con las pruebas, en las que puede haber que ir modificando y retocando lo desarrollado.

Cuando el sistema ya está desarrollado e implementado, se despliega y se fin pone al proyecto.

3.3.2 Gestión de desviaciones

Los plazos definidos en el diagrama de Gantt son los que ha de seguir el proyecto. No obstante, pueden darse desviaciones en los plazos de entrega, que si no se gestionan de forma correcta, encadenarían complicaciones en todo el desarrollo.

Para controlar el cumplimiento de la planificación, cada dos semanas se van a poner reuniones de seguimiento entre el cliente y el equipo, viendo a la imagen actual del sistema y pudiendo modificar la planificación en función de las necesidades del equipo y del estado del proyecto. En caso de que se hiciesen cambios en la planificación, éstos quedarían reflejados en un documento para llevar correctamente la gestión del proyecto.

4. ESTUDIO DE VIABILIDAD DEL SISTEMA

Esta tarea estudiará cómo de viable es la creación de este sistema, analizará el contexto en el que se va a desarrollar, viendo los posibles problemas que tiene la situación actual. Se analizarán distintas alternativas a la solución, en base a lo que el sistema necesita.

4.1 Establecimiento del alcance del sistema

4.1.1 Estudio de la solicitud

Este proyecto tiene como propósito plasmar en una aplicación diferentes tipos de eventos accesibles para personas con discapacidad auditiva y visual. De esta manera se expone de una forma más cercana al usuario este tipo de eventos para todas las personas que necesiten o requieran estas accesibilidades.

La aplicación que se quiere desarrollar se llama **ACA**. Es una agenda cultural donde se puede consultar los eventos que hay disponibles en determinadas ciudades, y con un servicio de accesibilidad concreto. El género de estos eventos es muy variado, ya que puede ser teatro, museo, cine u otro sin clasificación específica. Los servicios de accesibilidad que ofrece son subtítulos, audiodescripción, lenguaje de signos o una accesibilidad completa, que engloba las tres últimas.

El usuario puede consultar estos eventos en diferentes ciudades de España, todas aquellas que entren dentro del marco de ciudades que CESyA ofrece, que es el centro que va a porvenir los eventos de la aplicación.

Otro propósito de esta solicitud es que la aplicación sea multiplataforma, cubriendo las plataformas *Android* e *IOS*. Se pretende obtener un desarrollo que valga para ambos sistemas operativos, desempeñando estos aspectos:

- Desarrollar un solo código para el sistema
- Aproximar el desarrollo y mantenimiento de la aplicación a un único módulo
- Mejorar la calidad y escalabilidad que ofrecería un sistema nativo

4.1.2 Identificación del alcance del sistema

Este apartado relata la capacidad que tiene la aplicación de resolver las necesidades para las que ha sido planteado. ACA cuenta con dos tipos de usuarios:

- **Usuario registrado:** este tipo de usuario se ha registrado en la aplicación creándose una cuenta a partir de un formulario. Tiene acceso a la totalidad de servicios que ofrece ACA.
- **Usuario invitado:** este tipo de usuario no tiene la necesidad de crearse una cuenta. Puede acceder a la aplicación, pero cuenta con un acceso limitado a los servicios que ACA ofrece.

Para acceder a ACA, un usuario se encuentra en la ventana inicial tres opciones: iniciar sesión, registrarse o entrar sin registrarse. Identificados anteriormente los diferentes roles que tiene la

aplicación y en vista de la ventana inicial, se distinguen distintos alcances del sistema, en función del tipo de usuario:

4.1.2.1 Alcance del sistema: usuario registrado

Un usuario registrado es aquel que se crea una cuenta para acceder a en la aplicación. Por tanto, en la ventana inicial accede a la opción “¿Aún no te has registrado?” para acceder a registrarse en ACA. Dicha sección cuenta con un formulario, que está dividido en dos ventanas. Este formulario cuenta con estos campos:

- Nombre y apellidos (obligatorio)
- Correo electrónico (obligatorio)
- Nombre de usuario (obligatorio)
- Contraseña (obligatorio)
- Imagen de perfil (opcional)

Para llevar a cabo el registro, no puede existir su correo electrónico en la base de datos de ACA, ni otro nombre de usuario igual al suyo. Rellenando estos campos y siguiendo los requisitos necesarios, el usuario podrá finalmente crear su cuenta y guardar sus datos en las bases de datos de ACA, para poder acceder con su nombre de usuario y contraseña cuando lo desee. Una vez registrado, el usuario inicia sesión con sus datos registrados.

En cuanto el usuario se haya logueado, se encontrará con la ventana principal, donde por defecto se muestra visible un panel con las cuatro diferentes categorías de eventos que ofrece ACA. Puede navegar por cualquier categoría. Entrando en una de ellas, el usuario puede acceder a cualquier evento, viendo así información más detallada. Además de ver todos los eventos, el usuario tiene la posibilidad de marcarlos como “favorito”. Esta opción permite al usuario guardar eventos en otra sección, con el fin de saber cuáles le han llamado más la atención y poder acceder a ellos de una forma más rápida y directa en el momento que lo desee. Para marcar como favorito, el usuario tiene que tocar el icono de aspa en la esquina derecha inferior de cada evento.

Dentro de cada evento el usuario tiene la siguiente información adicional:

- Servicios de Accesibilidad: vienen indicados en formato de texto acompañados de un icono para hacer más fácil su identificación.
- Ubicación: contiene la dirección del lugar donde se celebra dicho evento, el teléfono y un acceso directo a su web.

ACA cuenta con eventos accesibles en casi todas las ciudades de España, por lo que el usuario tiene la opción de filtrar eventos por ciudad. Es posible que en la ciudad que el usuario busque no haya eventos, pero ACA se lo comunicará. Para agilizar esta búsqueda, ACA tiene una sección de “ciudades populares”, que son las más transitadas. Aún así, puede introducir el nombre de la ciudad si no está en esta sección.

Siempre que quiera filtrar por ciudad, también es necesario que indique un tipo de accesibilidad: subtítulo, audiodescripción, lenguaje de signos, o accesibilidad completa. No podrá aplicar estos filtros por separado, ni aplicarlos en distintas ciudades o distintos servicios accesibles a la vez.

Un método útil y accesible para el usuario es poder volver a la ventana principal desde cualquier punto de la aplicación, sin tener que pasar por el histórico de ventanas que ha visitado. Tocando el logo de la aplicación, que está centrado en la parte superior de casi todas las ventanas, se vuelve a la principal. Las ventanas que no tienen el logo, es porque tienen otra forma también sencilla para volver al inicio de la aplicación.

Los eventos que el usuario ha guardado como favoritos se encuentran en *Mi Archivo*. Esta pestaña, que está en la ventana principal, tiene listados los eventos que el usuario almacenó, separados por género. Desde esta pestaña puede también acceder a la información adicional de cada evento. También podrá desmarcarlo como favorito desde este mismo sitio como desde el panel de eventos de cada categoría (puede desmarcarlo siempre que lo haya marcado anteriormente). Si lo desmarca, desaparecerá de esta sección, y tiene la opción de volver a guardarlo cuando lo desee.

Para que el usuario esté al tanto de quiénes son los organizadores de esta iniciativa, existe en la ventana principal una sección llamada *CESyA*. Ésta muestra información que da a conocer a esta organización, acompañado de un gráfico con la relación de *CESyA* entre diferentes organismos.

El usuario, en toda ventana, puede acceder al menú lateral. Este menú facilita la navegación por la aplicación, porque cuenta con accesos directos a:

- *Mi Perfil*: ventana que contiene la información personal del usuario. Desde aquí puede modificar esta información, llevándole a la ventana *Configuración*. El usuario puede cerrar su sesión desde aquí.
- *Mi Archivo*: ventana donde están los eventos que ha marcado como favoritos
- *Acerca de*: ventana *CESyA*, para informarse acerca de la organización creadora de ACA
- *Contacta*: comunicación directa con *CESyA*, ya sea por correo electrónico o redes sociales
- *Cerrar sesión*: salir de su sesión y volver a la ventana inicial de la aplicación

La ventana *Configuración* contiene otro formulario donde el usuario podrá cambiar sus datos personales, modificando así la información de su cuenta.

Un usuario registrado puede entrar a la aplicación como usuario invitado, pero es requisito indispensable haber cerrado su sesión.

Para explicar de una forma más gráfica el alcance del sistema del usuario registrado, se ha diseñado un diagrama:

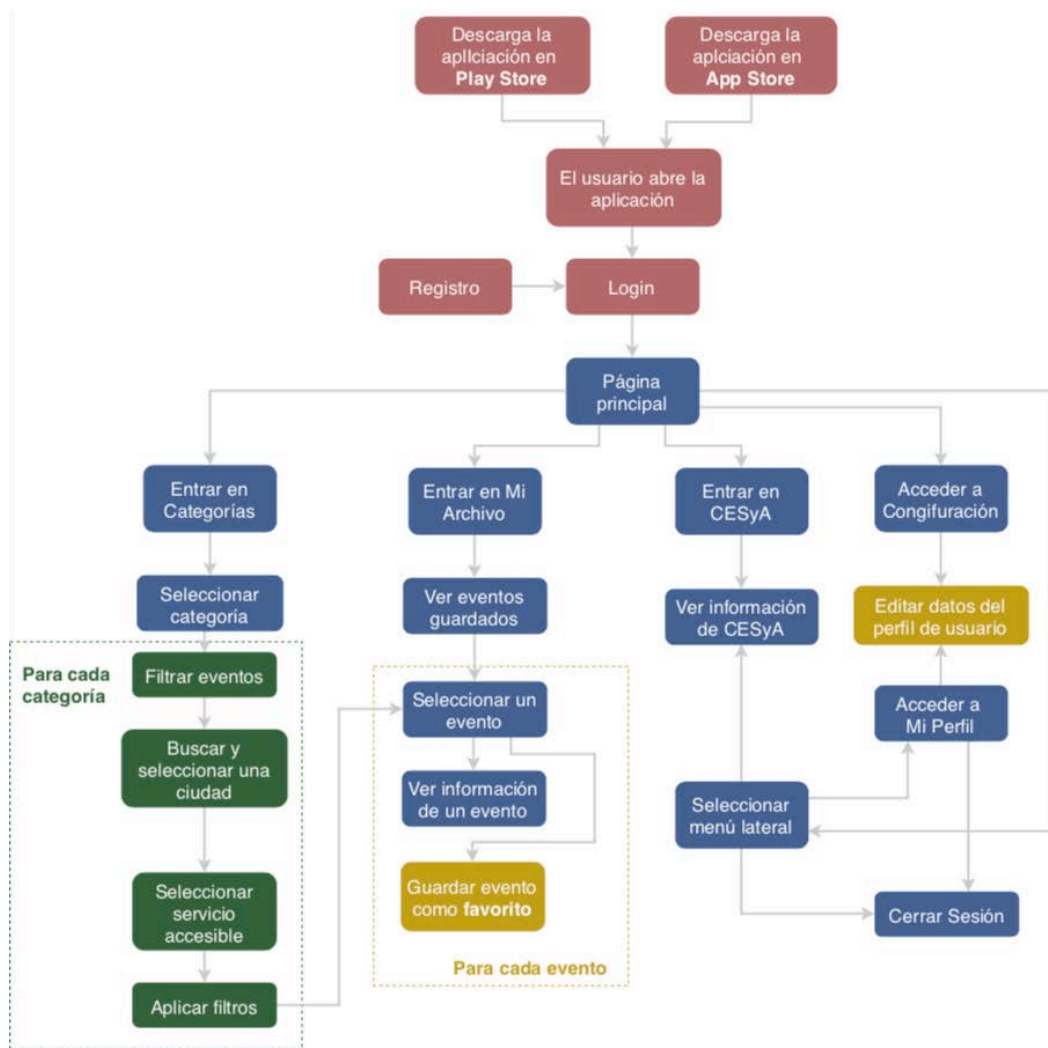


Ilustración 4: Alcance del sistema usuario registrado

4.1.2.1 Alcance del sistema: usuario invitado

Un usuario invitado es aquel que prefiere no crearse una cuenta para acceder a la aplicación. Este tipo de usuario implica tener restringidos ciertos servicios que ofrece ACA. No obstante, tiene los suficientes servicios en modo invitado como para poder disfrutar de la aplicación.

Cuando un usuario invitado entra a la aplicación, éste también puede acceder a la sección “Iniciar sesión”, pero como no tiene cuenta, no podrá continuar. Por tanto, el único camino viable para entrar es “Entrar sin registrarse”.

El usuario invitado tiene acceso al panel de las cuatro categorías de eventos. Puede acceder a ellas, viendo todos los eventos de cada una de ellas. Los eventos que se muestran por defecto son los disponibles en Madrid con accesibilidad completa, sin posibilidad de aplicar ningún filtro.

Si el usuario desea marcar algún evento como favorito, tiene visible el icono de aspa en cada evento. Si éste toca el icono, ACA le avisará que, para ello, debe registrarse. Lo mismo pasa con la pestaña *Mi Archivo*, en la ventana principal. Tiene acceso a ella, pero solo podrá ver un mensaje indicando que se registre para poder almacenar eventos.

Un método útil y accesible para el usuario es poder volver a la ventana principal desde cualquier punto de la aplicación, sin tener que pasar por el histórico de ventanas que ha visitado. Tocando el logo de la aplicación, que está centrado en la parte superior de casi todas las ventanas, se vuelve a la principal. Las ventanas que no tienen el logo, es porque tienen otra forma también sencilla para volver al inicio de la aplicación.

Para que el usuario esté al tanto de quiénes son los organizadores de esta iniciativa, existe en la ventana principal una sección llamada CESyA. Ésta muestra información que da a conocer a esta organización, acompañado de un gráfico con la relación de CESyA entre diferentes organismos.

El usuario, en toda ventana, puede acceder al menú lateral. Este menú facilita la navegación por la aplicación, porque cuenta con accesos directos a:

- *Registrarse*: el usuario tendrá opción de acceder al formulario del registro desde este menú
- *Mi Archivo*: ventana donde están los eventos que, si se registra, ha marcado como favorito
- *Acerca de*: ventana CESyA, para informarse acerca de la organización creadora de ACA
- *Contacta*: comunicación directa con CESyA, ya sea por correo electrónico o redes sociales

Para explicar de una forma más gráfica el alcance del sistema del usuario invitado, se ha diseñado un diagrama:

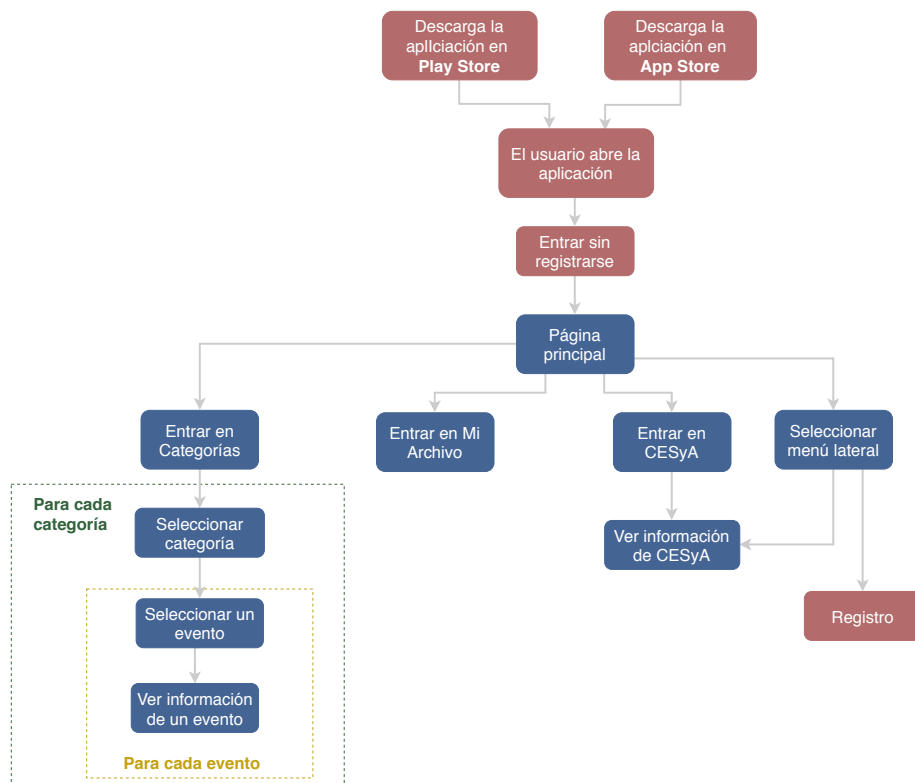


Ilustración 5: Alcance del sistema usuario invitado

4.1.3 Identificación de los interesados del sistema

En este apartado se analizan las personas implicadas en el sistema. Al tratarse de un TFG, los objetos de interés por los *stakeholders* será tanto la aplicación como todo el proyecto.

Los interesados, y por tanto, las personas que se verán afectadas con este proyecto son las siguientes:

STAKEHOLDER	DESCRIPCIÓN	OBJETO DE INTERÉS
Paula Laplana Martín	El alumno, encargado de desarrollar y documentar el proyecto	Proyecto
Israel González Carrasco	Tutor del proyecto, encargado de validar todo el desarrollo	Proyecto
Tribunal UC3M	Personal de la universidad encargado de aprobar el proyecto	Proyecto
Usuario final	Personas que finalmente utilicen la aplicación	Aplicación
CESyA	Encargado de mantener y administrar el sistema cuando se ponga en producción	Aplicación

Tabla 1: Interesados en el sistema

4.2 Marco Regulador

Una vez analizada la necesidad descrita por el usuario, será necesario estudiar las posibles restricciones que pudiesen afectar a la realización del proyecto. Dichas restricciones son de carácter económico, técnico, operativo y legal.

Este proyecto no cuenta con un gran importe inicial económico que limite el desarrollo, ya que todas las especificaciones necesarias para el desarrollo del proyecto son de carácter gratuito, salvando la estimación del presupuesto que se verá más adelante. Además, la propia organización que la ha creado no ha impuesto ninguna cantidad externa. Por tanto, inicialmente esta restricción puede contarse como una barrea de entrada, pero pequeña ya que no supone un gasto económico excesivo, fuera de lo común de un proyecto software. No obstante, más adelante se detallan pequeños gastos que ha ido suponiendo la realización de este desarrollo, concluyendo con el presupuesto del proyecto.

En cuanto a restricciones de carácter técnico y operativo, se encuentra el desconocimiento inicial del entorno de desarrollo de la aplicación. Esto supone tener que hacer un previo estudio de dicho entorno para familiarizarse con ello, además del propio desarrollo. Esta aplicación, aunque su desarrollo alcance todas las provincias de España, no tendrá la misma utilidad en las provincias más habitadas que en las que tengan un nivel de población y desarrollo inferior.

Respecto a restricciones legales, el proyecto debe estar sujeto al Reglamento de Protección de datos (RGPD/GDPR) (9) y la Ley Orgánica de Protección de Datos y Garantía de Derechos

Digitales (LOPD) (10). Bien es cierto que dicha aplicación no guardará cantidades masivas de datos, ya que solo será necesario la información personal de cada usuario.

Además, este proyecto debe contar, en caso de que las necesite, con las licencias de cada recurso que se use, desde librerías usadas en código, hasta la aceptación de facilitar la accesibilidad en los distintos establecimientos de eventos que ofrezca la aplicación.

Otra restricción de carácter legal está relacionada con los usuarios menores de edad. Se ha de tener especial cuidado en mantener un criterio de edades permitidas para cada evento que ofrezca la aplicación.

4.3 Estudio de la situación actual

Este apartado consta del estudio del entorno actual en el que se va a desarrollar el sistema. De esta manera se podrá identificar de una forma óptima el problema existente en los sistemas actuales, para encontrar solución a ello.

Para esta valoración se van a analizar aplicaciones ya existentes que puedan ser competencia, estudiando por qué son competidoras y las diferencias que existen entre ellas. En función de esta valoración se tendrán en cuenta las ventajas y desventajas que cuenta la aplicación ACA, pudiendo potenciarlas o hacerles frente.

4.3.1 Valoración de la situación actual

Para crear el contexto inicial de este estudio se expondrá el problema principal por el que se va a desarrollar la aplicación. También se va a detallar aplicaciones similares que cuentan con dicho problema, para comprenderlo de mejor manera.

4.3.1.1 Aplicaciones competidoras

Agenda Cultural Accesible -- ACAccesible

Esta aplicación está orientada a personas con discapacidad auditiva y visual, que quieran disfrutar de eventos con servicios accesibles. Cuenta con cuatro tipos de eventos diferentes: museo, teatro, cine y otros. La aplicación tiene un funcionamiento simple, ya que únicamente ofrece la información de los servicios, filtrados por ciudad y tipo de accesibilidad.

Dicho sistema está desarrollado de forma nativa, es decir, tiene un desarrollo para cada uno los diferentes sistemas operativos en los que se despliega (*Android* e *IOS*).

TOUR4all – PREDIF

TOUR4all es una aplicación para la organización de eventos accesibles. Aporta información fiable de eventos en distintas ciudades. Está destinada a usarla cuando los usuarios realizan turismo. Comparte todo tipo de eventos, ya sea desde encontrar

hoteles adaptados hasta transporte con silla de ruedas incluida. También cuenta con información acerca de eventos culturales.

Sin saber con certeza si se trata de una aplicación multiplataforma o nativa, ya que no es conocido el desarrollo de dicha aplicación, puede considerarse competencia de ACA, por englobar la misma información. No obstante, TOUR4all abarca mucha más información ya que el fin de la aplicación no es el mismo.

4.3.1.2 Problema de la situación actual

Actualmente, no existe ninguna aplicación de eventos accesibles que su único desarrollo sirva para varias plataformas. La aplicación *ACAcesible* no cuenta con un desarrollo multiplataforma, ya que se despliega por separado en *Android* y en *IOS*. Además, presenta un diseño con escasa funcionalidad. Por otro lado, la aplicación *TOUR4all* tampoco cubre todo el objetivo que se quiere cubrir con este proyecto, ya que está orientada con un fin distinto.

El problema actual, por tanto, es el coste elevado de mantener ambas aplicaciones para los diferentes tipos de sistemas operativos.

Es un problema en cuanto a coste económico, ya que de esta manera se están haciendo dos aplicaciones nativas totalmente independientes (aunque se trate de la misma), y de desarrollo, porque se está duplicando el esfuerzo y el trabajo. Además, a la hora de actualizar, mejorar o modificar el sistema, se ha de tener en cuenta dichos cambios en ambos desarrollos. Esto puede llegar a crear discrepancias entre las dos aplicaciones de cara al usuario, y no debería darse ya que la administración de la aplicación debe ser transparente al usuario final.

Otro problema no tan llamativo, es la poca funcionalidad de las aplicaciones que ofrecen información similar a la que se quiere desarrollar. En cuanto a *ACAcesible*, Lo único que proporciona es la información de los eventos. Aporta lo necesario, pero podría ser de una manera más óptima y completa. También presenta una usabilidad baja porque no sigue muchos cánones de diseño. Por ejemplo, no emplea un diseño atractivo para que sea agradable de usar para el usuario final, tampoco muestra que se haya pensado en el usuario a la hora de diseñar la interfaz, haciendo fácil y más usable cualquier servicio que ofrece la aplicación. Una aplicación de eventos accesibles debe ir de la mano de la usabilidad.

También se encuentra el problema de la escalabilidad. Una aplicación nativa, desarrollada para diferentes plataformas, presenta baja escalabilidad por tener que tratar varios sistemas para un mismo desarrollo. Puede llevar a perder calidad y es complicado la posibilidad de que crezca de manera fluida, por no tratarse de un sistema homogéneo.

4.4 Estudio de las alternativas de diseño

Una vez encontrado los problemas actuales, se van a analizar posibles opciones para solucionarlos:

- a) Realizar una aplicación híbrida, en la que un solo código/desarrollo sirva para todos los sistemas operativos que se desee. Además, mejorar el diseño y la usabilidad de la interfaz.

- b) Realizar una nueva aplicación nativa para cada sistema operativo, mejorando el diseño y la usabilidad de la interfaz.

Conociendo los problemas encontrados en la situación actual, se considera más completa e importante la primera opción, ya que abarca la solución tanto al problema del coste de mantenimiento como al problema de usabilidad y diseño de la interfaz. La otra posibilidad podría ser válida, pero no cubre todos los problemas actuales.

Por tanto, teniendo esta opción escogida, se analizan como alternativas los posibles entornos de desarrollo para realizar una aplicación multiplataforma:

4.4.1 Primera alternativa: Apache Córdoba

(11) Apache Córdoba es un marco de desarrollo móvil que permite usar lenguajes estándares web evitando los lenguajes de desarrollo nativo propios de cada sistema operativo. Estos lenguajes son HTML, JavaScript y CSS. Todos ellos son comunes para los sistemas operativos implicados en este proyecto (*Android e IOS*).

Este entorno permite crear líneas de trabajo multiplataforma o centrado en una sola plataforma. Ya que lo que se busca es la alternativa multiplataforma, esta opción permite dicho desarrollo a través de la interfaz de líneas de comandos. Dentro del directorio raíz del proyecto, crea diferentes subdirectorios, uno por cada plataforma y serán los que posteriormente se desplieguen en los diferentes sistemas operativos. Cada uno de estos subproyectos se modificará a medida que el proyecto base se modifique, sin necesidad de alterar todos.

(12) Apache Córdoba, por defecto, ofrece la misma interfaz para todas las plataformas. En caso de querer lo contrario, se puede lograr mediante los frameworks de CSS.

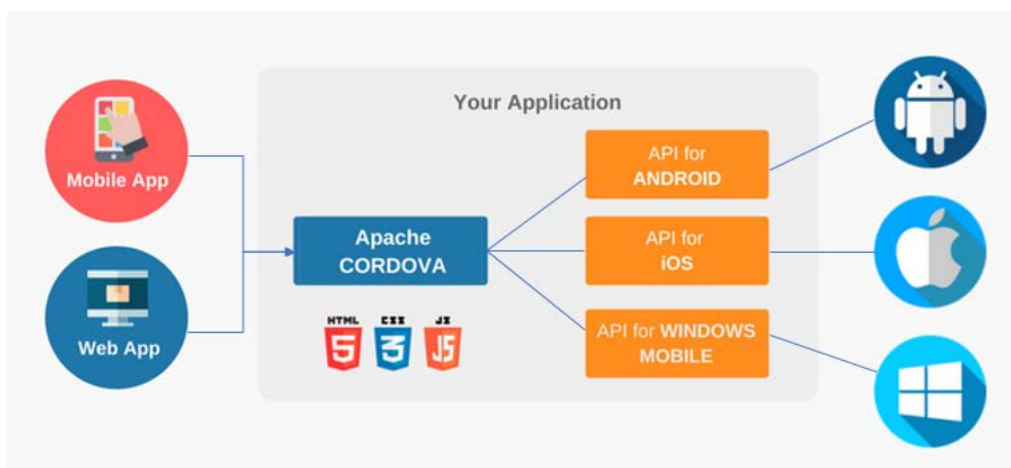


Ilustración 6: Arquitectura Apache Córdoba (13)

4.4.2 Segunda alternativa: Xamarin

(14) Esta herramienta de Microsoft permite desarrollar aplicaciones multiplataforma en lenguaje C# y XML para el formato y estilo. Convierte, mediante un único código base,

aplicaciones nativas para diferentes plataformas. Además, requiere conocimientos avanzados de .NET, *Android* e *IOS* para su desarrollo.

Este entorno de desarrollo ofrece un buen rendimiento y funcionalidad, ya que se compila a código nativo y accede a las APIs nativas mediante envoltorios de C#. Esto hace que de apariencia parezca que se trate de aplicaciones totalmente nativas, pero con un desarrollo y mantenimiento mucho más óptimo.

Gracias a que Xamarin accede a las APIs nativas, permite crear interfaces diferentes para las diferentes plataformas.

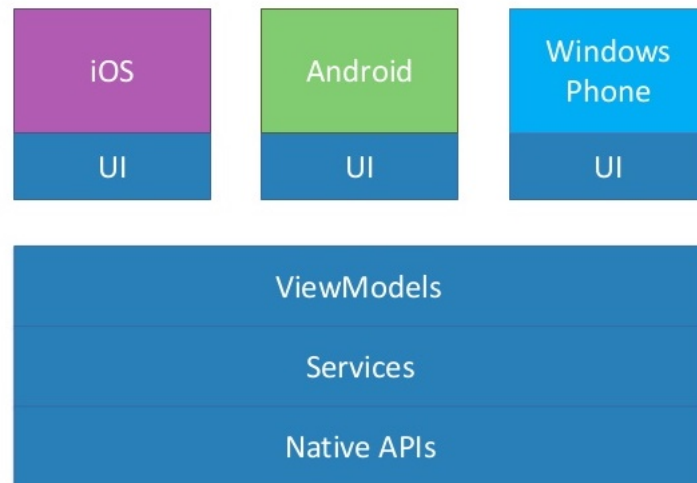


Ilustración 7: Arquitectura Xamarin (15)

4.5 Valoración de las alternativas de diseño

(16) A continuación, se muestra la valoración de cada alternativa encontrada. Se estudiarán sus ventajas y desventajas para este proyecto. Posteriormente se va a escoger una de ellas para llevar a cabo la solución final.

4.5.1 Apache Córdova

Las ventajas que presenta este marco de desarrollo son las siguientes:

- La interfaz de usuario que proporciona es igual para todas las plataformas. ACA requiere que todas las versiones contengan el mismo aspecto.
- Se cuenta con un conocimiento avanzado de los lenguajes estándares que emplea (HTML, CSS y JavaScript).
- Este marco de desarrollo móvil se implementa mediante líneas de comandos. Al desarrollarse este proyecto en el sistema operativo MacOS, el administrador ya está familiarizado con la terminal. En caso de ser con otro sistema operativo, podría tratarse de una barrera de entrada.
- Siguiendo el punto anterior, no necesita la instalación de un software adicional. Sí requiere de instalaciones del SDK de Android, pero se implementa en la terminal del propio ordenador.
- No requiere de conocimientos avanzados de Android e IOS, las plataformas finales del proyecto.

- Se trata de un entorno opensource, con la posibilidad de obtener plugins de apoyo al desarrollo.
- Soporta los sistemas operativos necesarios para este proyecto.

Apache Córdoba también cuenta con estas desventajas:

- Desconocimiento del propio entorno de desarrollo. Aunque su implementación sea mediante la terminal, implica tener que afianzarse a dicho marco con sus propios comandos.
- El mantenimiento de este sistema en el futuro implica que se tenga conocimientos de la terminal del ordenador donde se siga desarrollando.
- La interfaz de usuario que modela no presenta ningún aspecto nativo. En caso de necesitarse, sería posible con el framework de CSS.
- Presenta un rendimiento lento en antiguos dispositivos, independiente a la plataforma en la que se despliegue.
- Cuenta con un tiempo de arranque inicial de la aplicación de baja velocidad.
- Para conseguir un rendimiento mayor se necesitan plugins adicionales de Multithreading.

4.5.2 Xamarin

Este entorno de desarrollo cuenta con estas ventajas:

- Tiene la capacidad de proporcionar interfaces nativas para cada plataforma. Las características nativas son compatibles y pueden ser implementadas.
- Presenta una alta capacidad de respuesta de la interfaz de usuario.
- Emplea ejecución multithread consiguiendo un alto rendimiento en todas las plataformas.
- Tiene acceso completo a las APIs de cada sistema operativo.
- Cuenta con un tiempo de arranque inicial de alta velocidad.
- Soporta los sistemas operativos necesarios para este proyecto.

Xamarin presenta, además, estas desventajas:

- Desconocimiento de los lenguajes de programación .NET y C#.
- Requiere un conocimiento avanzado de XML, Android e IOS para desarrollar una aplicación.
- Requiere instalación de software adicional, Visual Studio. Presenta una compatibilidad compleja con MacOS, el sistema operativo donde se va a desarrollar el proyecto.
- El mantenimiento de este proyecto en un futuro implica que tenga el software necesario instalado.
- Es un entorno de desarrollo opensource, pero con diferentes licencias que podrían suponer cierto coste económico.

4.6 Selección de la alternativa de diseño

Tras el estudio de las alternativas se ha visto que este proyecto sería viable con ambas opciones. Sin embargo, hay que escoger una de ellas para llevar a cabo el desarrollo.

Para elegir la alternativa final, se van a seguir ciertos criterios ponderados de forma numérica. La alternativa que cuente con una puntuación más alta será la elegida.

La ponderación escogida oscila del 1 al 5, siendo 1 la menor puntuación y 5 la máxima. Los criterios a seguir, con sus puntuaciones en cada alternativa, son las siguientes:

CRITERIO	Apache Córdoba	Xamarin
Rendimiento	4	5
Coste económico *	5	3
Aprendizaje	2	4
Familiarización con el entorno	5	2
Mantenimiento	4	4
Seguridad	5	5
Soporte	4	2
Escalabilidad	5	5
TOTAL	34	30

Tabla 2: Criterios selección alternativa

* Este criterio se pondera de manera inversa, puesto que un menor coste económico implicaría la mayor ponderación y un elevado coste económico contaría con la menor puntuación.

Viendo el resultado, la primera alternativa cuenta con una ponderación superior. Bien es cierto que Xamarin aportaría un alto rendimiento, factor muy importante en un desarrollo software. Pero el resto de factores, contando con la recomendación del tutor de este proyecto, hacen que la más valorada, y por tanto la escogida, sea Apache Córdoba.

4.7 Entorno socio-económico

Para llevar a cabo este proyecto, sabiendo ya en qué entorno se va a desarrollar, se ha de estimar ciertos factores para la realización del mismo. Hay que tener en cuenta la estimación de horas de trabajo, los costes que supondrá o el personal implicado que desarrollará el sistema, entre otras.

Hay que tener en cuenta que este proyecto es un Trabajo de Fin de Grado, por lo que el único desarrollador del sistema será el propio alumno, ya que los TFG son de carácter individual.

4.7.1 Estimación de costes

Para estimar los costes, se analiza el equipo, componentes y material que se va a necesitar.

4.7.1.1 Componentes Software

Este proyecto requiere de los siguientes componentes software:

- Navegador web: Safari, Google Chrome y Firefox
- Editor de texto: Komodo 11
- Entorno de compilación: terminal del ordenador, Apache Córdoba
- Entorno de ejecución: Android Studio, Xcode y terminal del ordenador
- Plataforma para la gestión de datos: Firebase de Google
- Base de Datos: Realtime Database de Firebase de Google
- Lector de documentación: Vista Previa de MacOS
- Diseño de diagramas UML: draw.io (online)
- Diseño de planificación: SmartSheet (online)

COMPONENTE SOFTWARE	PRECIO
Navegadores web	0,00€
Komodo 11	0,00€
Apache Córdoba	0,00€
Android Studio	0,00€
XCode	0,00€
Firebase	0,00€
Draw.io	0,00€
SmartSheet	0,00€
TOTAL	0,00€

Tabla 3: Estimación costes Software

Se observa que este proyecto se va a desarrollar con software opensource, es decir, totalmente gratis. Por lo que por esta parte no se estima ningún coste económico.

4.7.1.2 Equipo Hardware

Este proyecto requiere de los siguientes componentes hardware:

- Ordenador portátil: MacBook Pro 2016 de Apple
- Dispositivo auxiliar de ayuda: Ipad 2018 de Apple
- Dispositivo para control de plataforma Android: Xiaomi Mi A1 - Android 9

- Dispositivo para control de plataforma IOS: iPhone 7 – IOS 12
- Monitor de ayuda: Pantalla LG FALTRON L1952S
- Disco duro externo: WD My Passport 1TB
- Hub para puertos USB HP tipo C
- Impresora: HP ENVY 2000 Series

Muchos de los dispositivos necesarios ya se disponen, por lo que no añaden ningún coste al proyecto. Bien es cierto que se tienen que comprar ciertos materiales para desarrollarlo:

DISPOSITIVO	PRECIO	COMENTARIOS
Ordenador portátil MacBook Pro 2016 de Apple	1.355,89€	Aplicado un 10% de descuento por ser estudiante
Disco duro externo WD My Passport 1TB	69,90€	

Tabla 4: Estimación costes Hardware

Haciendo un cálculo total de los costes de equipo que supondrá este proyecto, se estima este coste:

MATERIAL	PRECIO
Componentes Software	0,00€
Equipo Hardware	1.355,89€
	69,90€
TOTAL	1.425,79€

Tabla 5: Estimación total de costes

El total que se estima de costes para este proyecto es de 1.425,79€.

4.7.2 Estimación de personal

Como se ha mencionado anteriormente, este proyecto es de carácter individual por tratarse de un TFG. Esto significa que es el propio alumno es el que desempeña todos los roles necesarios para el desarrollo de un proyecto software.

A continuación, se muestra los cargos necesarios para este desarrollo:

CARGO	ENCARGADO	HORAS
Jefe de proyecto	Paula Laplana Martín	1.480
Analista	Paula Laplana Martín	152
Diseñador	Paula Laplana Martín	160
Programador	Paula Laplana Martín	456
Gestión de calidad	Paula Laplana Martín	136
Gestión de pruebas	Paula Laplana Martín	248

Tabla 6: Estimación de personal

En cuanto al personal, se tienen en cuenta las horas que realizará cada encargado en base a su categoría. Además, se estiman jornadas laborales de 8h.

4.7.3 Estimación de presupuesto

La estimación del presupuesto se realizará en base a la duración del proyecto, estimando las horas que suponen cada tarea, y al personal implicado.

4.7.3.1 Sueldo del personal

TAREA	RESPONSABLE	DIAS	HORAS POR ENCARGADO	€/h
Gestión	Jefe del proyecto	185	1.480	18.114,58€
Planificación	Jefe del proyecto	11	88	1.077,08€
Estudio de viabilidad	Jefe del proyecto	17	136	1.664,58€
	Analista	17	136	1.644,58€
Estimación	Jefe del proyecto	39	312	3.818,75€
Construcción	Jefe del proyecto	57	456	5.581,25€
	Analista	35	280	3.427,08€
	Diseñador	22	160	1.958,33€
Validación	Jefe de Proyecto	31	248	3.035,41€
	Diseñador	14	112	1.370,83€
	Programador	17	136	1.664,58€

Tabla 7: Estimación €/horas por tarea

CARGO	ENCARGADO	HORAS	€
Jefe de proyecto	Paula Laplana Martín	1.480	18.114,58€
Analista	Paula Laplana Martín	152	1.860,41€
Diseñador	Paula Laplana Martín	160	1.958,33€
Programador	Paula Laplana Martín	456	5.581,25€
Gestión de calidad	Paula Laplana Martín	136	1.664,58€
Gestión de pruebas	Paula Laplana Martín	248	3.035,41€
TOTAL			32.214,56€

Tabla 8: Estimación sueldo personal

En un equipo real, el coste de personal sería de 32.214,46€, siendo el total del sueldo de cada encargado. Ya que este proyecto es un trabajo académico, no hay coste de personal.

El coste económico de cada encargado se ha calculado multiplicando el número de horas que va a destinar al proyecto por el coste de 1 hora de trabajo.

4.7.3.2 Equipo informático

Como se ha estimado anteriormente se requiere de 1.425,79€ para la inversión en equipo y material necesario para el desarrollo. Teniendo en cuenta que este equipo informático se va a amortizar en 4 años, y que la duración de este proyecto es de 6 meses, el presupuesto en cuanto al equipo informático que le corresponde a este proyecto sería en siguiente:

$$(1.425,79 / (4 * 12)) * 6 = 142,57 \text{ €}$$

4.7.3.3 Viajes y dietas

Para el desarrollo de este proyecto será necesario realizar algunos viajes para mantener las reuniones estimadas con el cliente. Estas reuniones se realizan con el tutor en la universidad una vez al mes, ya que para consultas más específicas se comunican mediante correo electrónico. Por tanto, teniendo en cuenta los 7 meses aproximados de duración del proyecto:

- **Viajes:** se estima un gasto en gasolina de 12€ por cada reunión, habiendo 1 reunión por mes y teniendo en cuenta a todas las personas del equipo
- **Dietas:** se estima un gasto de 10€ por persona en cada reunión

ELEMENTO	GASTO MES /persona	GASTO MES TOTAL	TOTAL PROYECTO
Viajes	2,00€	12,00€	72,00€
Dietas	10,00€	60,00€	360,00€
TOTAL			432,00 €

Tabla 9: Estimación viajes y dietas

4.7.3.4 Costes indirectos

Dentro del proyecto pueden darse costes indirectos, como el gasto de la luz del lugar de trabajo, servicios de internet y telefonía, agua, etc. La estimación de estos costes indirectos es la siguiente, teniendo en cuenta que la duración del proyecto es de 6 meses:

ESTIMACIÓN	GASTO POR MES	TOTAL PROYECTO
Alquiler oficina	1900,00€	11.400,00€
Luz	200,00€	1.200,00€
Agua	50,00€	300,00€
Gas	80,00€	480,00€
Línea de internet	50,00€	300,00€
Líneas móviles	120,00€	720,00€
TOTAL		14.400,00 €

El gasto en costes indirectos se eleva a 14.400,00€.

Dado a que es un trabajo académico no se estima necesario tener que invertir en un lugar de trabajo o derivados que impliquen costes indirectos. Por tanto, se no se destinará nada al gasto en costes indirectos.

4.7.3.5 Margen de riesgo y beneficio

ELEMENTO	PORCENTAJE A APLICAR	VALOR BRUTO	VALOR TOTAL
Sueldo de personal	0%	32.214,56€	32.214,56€
Amortización equipo informático	0%	142,57€	142,57€
Viajes y dietas	0%	432,00€	432,00€
Costes indirectos	0%	14.400,00€	14.400,00€
Coste total proyecto	0%	47.189,13€	47.189,13€
Margen de riesgo	+25%	11.797,28€	58.986,41€
Margen de beneficio	+15%	8.847,96€	67.834,37€
IVA	+21%	14.245,21€	82.079,58€
TOTAL PRESUPUESTO			82.079,58€

Tabla 10: Margen de riesgo y beneficio

4.7.3.6 Estimación total del presupuesto del proyecto

Se estima finalmente que el presupuesto de este proyecto se eleva a 82.079,58€.

5. ANALISIS DEL SISTEMA

Este apartado tiene el objetivo de obtener una especificación detallada del sistema que se va a construir, sirviendo de base para el posterior diseño.

Presenta la definición formal de las necesidades del usuario, al igual que las especificaciones de las normas y estándares que la aplicación debe seguir. Además, esta tarea estudiará los subsistemas en los que se divide el sistema completo, logrando cohesión entre los mismos. También se definirán las interfaces de usuario que van a definir la aplicación, trazándolas con las acciones que el usuario puede realizar sobre ella.

Finalmente, en este apartado se estudiará el modelo de datos que sigue el sistema.

5.1 Definición del sistema

La aplicación ACA se define como una agenda cultural que proporciona eventos accesibles para personas con discapacidad auditiva y visual. Este sistema provee al usuario de eventos que cuenten con eventos accesibles en una determinada ciudad con un determinado servicio de accesibilidad.

El problema que va a cubrir este proyecto es el coste de mantenimiento de una aplicación nativa. Es por ello que la solución a dicho problema será crear una nueva aplicación multiplataforma cubriéndola de funcionalidad y estableciendo un diseño que cumpla los cánones de interfaces de usuarios.

5.1.1 Especificación de normas y estándares

Para la correcta realización de este proyecto, se van a seguir ciertos estándares y normas consiguiendo un desarrollo de calidad.

La documentación de este proyecto sigue una adaptación de la metodología Métrica V3 [*], que contiene los aspectos más fundamentales que implican al proyecto.

Para modelar la solución de este desarrollo se utilizará el estándar UML.

Este proyecto va a implicar acciones que conllevan el almacenamiento de datos personales. Esta información se almacenará según la Ley Orgánica de Protección de Datos y Garantía de Derechos Digitales (LOPD) (10) y el Reglamento General de Protección de Datos (RGPD/GDPR) (9).

5.1.2 Especificación del entorno operacional

Las restricciones generales que va a seguir el entorno operacional del sistema es el siguiente:

- El proyecto se creará a partir del marco de desarrollo Apache Córdoba.
- La aplicación ha de ser multiplataforma, cubriendo los sistemas operativos *Android* e *iOS*.
- Los lenguajes de programación seguidos para el desarrollo serán HTML, CSS y JavaScript.

- La aplicación tomará de servidor inicial el ordenador del alumno.
- La aplicación utilizará una base de datos en tiempo real para que se pueda desplegar desde cualquier cliente.
- El sistema requiere tener conexión a internet para poder iniciarse.
- La base de datos que lleve el proyecto gestionará los usuarios de la aplicación.
- El sistema realizará peticiones a servicios web para gestionar los eventos de la aplicación.
- El idioma empleado para el proyecto será el español.
- El servidor que mantiene a la aplicación en cada cliente será la plataforma que provea al usuario la propia aplicación (*App Store o Play Store*)

5.2 Especificación de casos de uso

Para definir las acciones que el usuario puede realizar en la aplicación, se especificarán los posibles casos de uso que permite el sistema.

La aplicación presenta diferentes actores, dependiendo de las acciones que tenga permitido hacer sobre el sistema. Se detallarán, en forma de diagrama, los casos de uso que desarrollará cada actor. Además, en formato tabular, se especificará cada caso de uso clasificado según su funcionalidad.

Los actores que presenta el sistema son los siguientes:

- **Usuario registrado:** usuario capaz de registrarse y loguearse en la aplicación, con permiso total para acceder a todos los servicios que ofrece el sistema
- **Usuario invitado:** usuario que entra en la aplicación sin crearse una cuenta en el sistema. Tiene acceso restringido a los servicios que ofrece la aplicación

Estos actores interactúan de forma diferente con el sistema. Se van a definir diferentes escenarios para ambos actores, en función de la funcionalidad de cada caso de uso. Los distintos escenarios son los siguientes:

- **Escenario 1:** interacción entre los actores y los casos de uso de acceso a la aplicación
- **Escenario 2:** interacción entre los actores y los casos de uso de navegación por la aplicación
- **Escenario 3:** interacción entre los actores y los casos de uso de filtro de eventos
- **Escenario 4:** interacción entre los actores y los casos de uso de almacenamiento de datos

5.2.1 Diagramas de casos de uso

A continuación, se muestra la definición de los diagramas de casos de uso para cada escenario:

5.2.1.1 Escenario 1: Acceso a la aplicación

El diagrama que se va a mostrar identifica cómo interactúan los actores con los casos de uso de acceso a la aplicación:

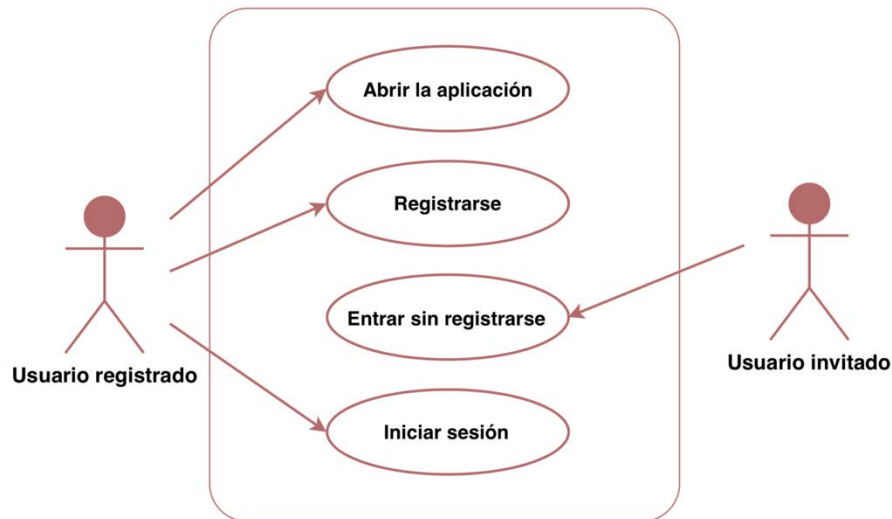


Ilustración 8: Diagrama CU Acceso a la aplicación

5.2.1.2 Escenario 2: Navegación por la aplicación

Los diagramas que se va a mostrar identifican cómo interactúan los actores con los casos de uso de navegación por la aplicación:

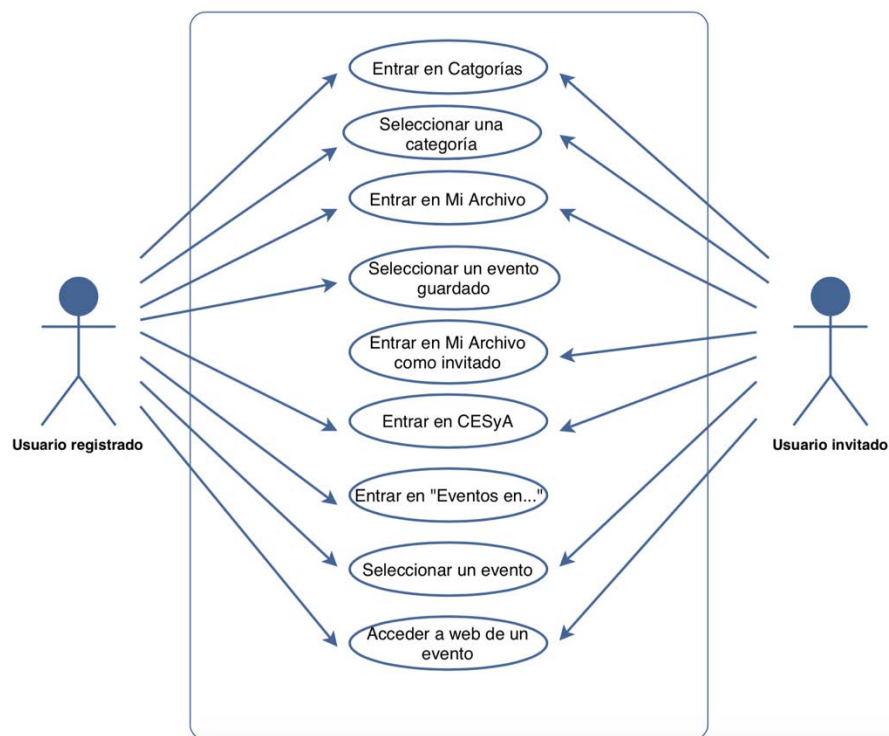


Ilustración 9: Diagrama CU Navegación por la aplicación I

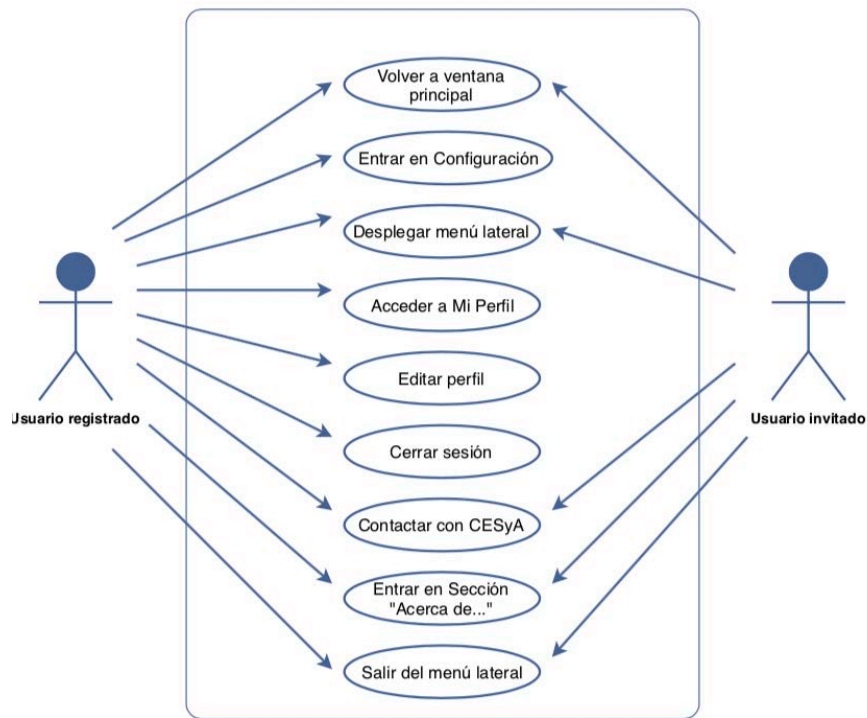


Ilustración 10: : Diagrama CU Navegación por la aplicación II

5.2.1.3 Escenario 3: Filtro de eventos

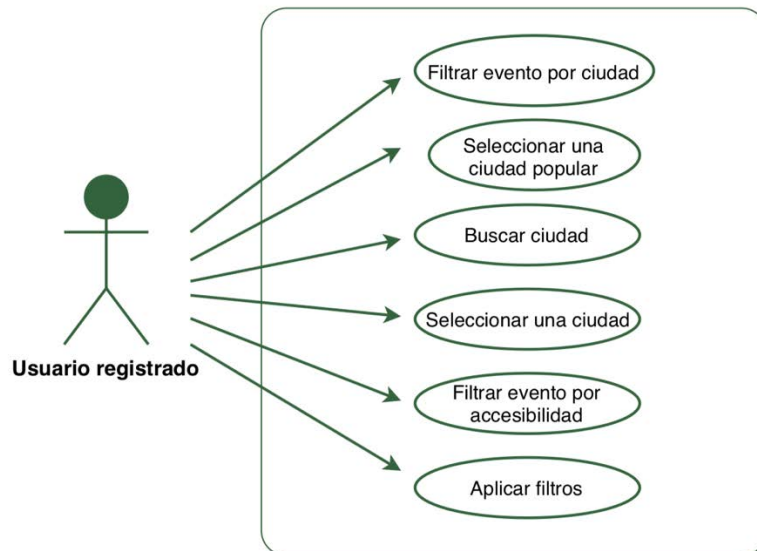


Ilustración 11: Diagrama CU Filtro de eventos

5.2.1.4 Escenario 4: Almacenamiento de datos

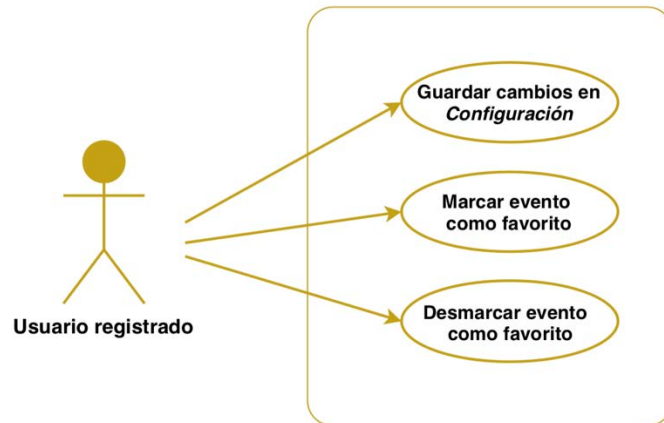


Ilustración 12: Diagrama CU Almacenamiento de datos

5.2.2 Definición de casos de uso

Cada caso de uso se va a detallar siguiendo la descripción de esta tabla:

ID	CU-XX
Caso de uso	
Actor	
Objetivo	
Precondiciones	
Postcondiciones	
Notas	

Tabla 11: Definición de casos de uso

Los campos de la tabla tienen la siguiente definición:

- **ID:** valor alfanumérico que identifica de forma única al caso de uso. Sigue este formato:
 - CU: siglas de *Caso de Uso*
 - XX: dígito numérico de dos cifras
 - $XX \in [0,99]$
- **Caso de uso:** nombre descriptivo del caso de uso
- **Actor:** persona o personas que interactúa con el caso de uso

- Actor ∈ [Usuario registrado, Usuario invitado]
- **Objetivo:** funcionalidad que se quiere conseguir con este caso de uso
- **Precondiciones:** condiciones que se deben cumplir para que se realice el caso de uso
- **Postcondiciones:** estado posterior a la realización del caso de uso
- **Notas:** aclaraciones de la realización del caso de uso. Este campo sólo estará en los casos de uso que lo necesiten

A continuación, se va a detallar cada caso de uso según su funcionalidad:

5.2.2.1 Casos de uso de acceso a la aplicación

ID	CU-00
Caso de uso	Abrir la aplicación
Actor	<ul style="list-style-type: none"> • Usuario registrado • Usuario invitado
Objetivo	El usuario podrá tener acceso a ACA
Precondiciones	<ul style="list-style-type: none"> • El usuario debe tener un sistema operativo compatible y descargarse la aplicación • El usuario deberá abrir la aplicación desde su dispositivo
Postcondiciones	EL usuario tendrá acceso a la pantalla principal de la aplicación

Tabla 12: Abrir la aplicación CU-00

ID	CU-01
Caso de uso	Entrar sin registrarse
Actor	<ul style="list-style-type: none"> • Usuario registrado • Usuario invitado
Objetivo	El usuario podrá acceder a la aplicación sin registrarse
Precondiciones	<ul style="list-style-type: none"> • El usuario debe estar dentro de la aplicación • El usuario puede no tener cuenta registrada en ACA
Postcondiciones	EL usuario podrá acceder a la pantalla principal de la aplicación, con las limitaciones de usuario invitado

Tabla 13: Entrar sin registrarse CU-01

ID CU-02

Caso de uso	Registrarse
Actor	<ul style="list-style-type: none"> • Usuario registrado • Usuario invitado
Objetivo	<p>El usuario se registra aportando los siguientes datos:</p> <ul style="list-style-type: none"> • Nombre y apellidos • Email • Nombre de usuario • Contraseña • Imagen de perfil
Precondiciones	<ul style="list-style-type: none"> • El usuario debe estar dentro de la aplicación • El usuario debe disponer de un correo electrónico válido • El correo electrónico del usuario no puede estar ya registrado en ACA • El nombre de usuario escogido no debe de existir en ACA • La contraseña debe tener ser alfanumérica con una longitud de 8 caracteres • Todos los campos son obligatorios salvo la imagen de perfil, que es opcional
Postcondiciones	EL usuario dispondrá de una cuenta personal en ACA, con la que podrá acceder a la misma y tener acceso a espacios acotados a usuarios registrados
Notas	Un usuario registrado puede acceder a la ventana “Registrarse”, pero para poder registrarse deberá usar otros datos diferentes a los de su perfil

Tabla 14: Registrarse CU-02

ID CU-03

Caso de uso	Iniciar sesión
Actor	Usuario registrado
Objetivo	El usuario inicia sesión para poder entrar a la aplicación
Precondiciones	<ul style="list-style-type: none"> • El usuario debe estar registrado • El usuario debe introducir de forma correcta su nombre de usuario y contraseña que escogió al crear su cuenta

Postcondiciones	EL usuario podrá acceder a su cuenta y navegar por todos los niveles la aplicación
Notas	Un usuario invitado puede acceder a la ventana “Iniciar sesión” pero no podrá iniciar sesión ya que no tiene cuenta en ACA

Tabla 15: Iniciar sesión CU-03

5.2.2.2 Casos de uso de navegación por la aplicación

ID CU-04

Caso de uso	Entrar en Categorías
Actor	<ul style="list-style-type: none"> • Usuario registrado • Usuario invitado
Objetivo	El usuario podrá entrar en la sección Categorías
Precondiciones	El usuario debe haber entrado a la aplicación
Postcondiciones	EL usuario tendrá acceso a las cuatro diferentes categorías

Tabla 16: Entrar en “Categorías” CU-04

ID CU-05

Caso de uso	Seleccionar una categoría
Actor	<ul style="list-style-type: none"> • Usuario registrado • Usuario invitado
Objetivo	El usuario podrá elegir entre las cuatro categorías de eventos
Precondiciones	El usuario debe haber entrado a la aplicación y estar en la ventana “Categorías”
Postcondiciones	EL usuario tendrá acceso al panel de eventos de la categoría que haya seleccionado

Tabla 17: Seleccionar una categoría CU-05

ID CU-06

Caso de uso	Entrar en Mi Archivo
Actor	Usuario registrado
Objetivo	El usuario podrá entrar en la sección Mi archivo, donde se almacenan listados los eventos que ha guardado como favoritos
Precondiciones	El usuario debe haberse registrado y logueado en la aplicación
Postcondiciones	EL usuario tendrá acceso a los eventos de todas las categorías que ha guardado de la aplicación

*Tabla 18: Entrar en Mi Archivo CU-06***ID CU-07**

Caso de uso	Seleccionar un evento guardado
Actor	Usuario registrado
Objetivo	El usuario podrá acceder a la información adicional del evento que ha marcado como favorito
Precondiciones	<ul style="list-style-type: none"> • El usuario debe haberse registrado y logueado en la aplicación • El usuario debe estar en la ventana Mi archivo • El usuario debe haber guardado anteriormente un evento
Postcondiciones	El usuario tendrá un acceso directo a los eventos que almacenó en Mi Archivo

*Tabla 19: Seleccionar un evento guardado CU-07***ID CU-08**

Caso de uso	Entrar en Mi Archivo como invitado
Actor	Usuario invitado
Objetivo	El usuario podrá entrar en la sección Mi archivo, limitado a las acciones de usuario invitado

Precondiciones	<ul style="list-style-type: none"> • El usuario debe haber entrado en la aplicación • El usuario no se ha registrado ni logueado en la aplicación • El usuario debe estar en la ventana principal
Postcondiciones	EL usuario estará en la ventana “Mi Archivo”, limitado únicamente a ver el espacio disponible para eventos guardados si se registra
Notas	Un usuario registrado podría acceder a “Mi Archivo” como invitado, siempre y cuando haya cerrado sesión con su cuenta y haya entrado sin registrarse a la aplicación

Tabla 20: Seleccionar un evento guardado CU-08

ID CU-09

Caso de uso	Entrar en CESyA
Actor	<ul style="list-style-type: none"> • Usuario registrado • Usuario no registrado
Objetivo	El usuario podrá seleccionar la ventana CESyA, donde dispone de información acerca de CESyA
Precondiciones	El usuario debe haber entrado a la aplicación
Postcondiciones	EL usuario tendrá acceso a información adicional de la organización creadora de ACA

Tabla 21: Entrar en CESyA CU-09

ID CU-10

Caso de uso	Entrar en “Eventos en...”
Actor	Usuario registrado
Objetivo	El usuario podrá seleccionar la ventana “Eventos en...” para aplicar diferentes filtros sobre los eventos
Precondiciones	El usuario debe haberse registrado y logueado en la aplicación y estar en una categoría concreta
Postcondiciones	EL usuario tendrá acceso a información adicional de la organización creadora de ACA
Notas	El usuario invitado verá esta pestaña con los eventos disponibles filtrados por “Madrid” y “Accesibilidad completa”, sin opción a cambiarlos

Tabla 22: Entrar en “Eventos en...” CU-10

ID CU-11

Caso de uso	Seleccionar un evento
Actor	<ul style="list-style-type: none"> • Usuario registrado • Usuario no registrado
Objetivo	El usuario podrá acceder a la información de un evento concreto dentro de la categoría seleccionada
Precondiciones	<ul style="list-style-type: none"> • El usuario debe haber entrado a la aplicación y estar dentro de una categoría • Debe haber eventos disponibles en esa categoría
Postcondiciones	EL usuario tendrá la información completa del evento seleccionado

*Tabla 23: Seleccionar un evento CU-11***ID CU-12**

Caso de uso	Acceder a web de un evento
Actor	<ul style="list-style-type: none"> • Usuario registrado • Usuario no registrado
Objetivo	El usuario podrá acceder a web del sitio que presenta el evento para obtener más información
Precondiciones	<ul style="list-style-type: none"> • El usuario debe haber entrado a la aplicación y estar dentro de una categoría • Debe haber eventos disponibles en esa categoría • El usuario debe haber seleccionado un evento
Postcondiciones	EL usuario estará en la web oficial del establecimiento que proporcione dicho evento

*Tabla 24: Acceder a web de un evento CU-12***ID CU-13**

Caso de uso	Volver a ventana principal
Actor	<ul style="list-style-type: none"> • Usuario registrado • Usuario no registrado
Objetivo	El usuario podrá tocar en el logo de la aplicación, colocado en la parte superior de la pantalla, para volver a la ventana principal

Precondiciones	<ul style="list-style-type: none"> El usuario debe haber entrado a la aplicación y no estar en la ventana principal La pantalla en la que se encuentre el usuario debe tener el logo de la aplicación
Postcondiciones	EL usuario podrá volver a la ventana principal desde cualquier otra ventana que contenga el logo de la aplicación
Notas	El logo que aparece en las pantallas de inicio de sesión, registro y pantalla inicial y principal no retornan a la pantalla principal ya que son anteriores a ésta o están en ella

Tabla 25: Volver a ventana principal CU-13

ID CU-14

Caso de uso	Volver a ventana anterior
Actor	<ul style="list-style-type: none"> Usuario registrado Usuario no registrado
Objetivo	El usuario podrá volver a la ventana anterior
Precondiciones	<ul style="list-style-type: none"> El usuario debe haber entrado a la aplicación La ventana en la que se encuentre el usuario debe tener el icono de “volver atrás”
Postcondiciones	El usuario tendrá acceso a la ventana en la que estaba anteriormente

Tabla 26: Volver a ventana anterior CU-14

ID CU-15

Caso de uso	Entrar en Configuración
Actor	Usuario registrado
Objetivo	El usuario podrá acceder a la ventana Configuración para editar los datos de su perfil
Precondiciones	El usuario debe estar registrado y logueado en la aplicación
Postcondiciones	EL usuario tendrá acceso a cambiar sus datos de perfil

Tabla 16: Entrar en Configuración CU-15

ID CU-16

Caso de uso	Desplegar el menú lateral
Actor	<ul style="list-style-type: none"> • Usuario registrado • Usuario no registrado
Objetivo	El usuario podrá acceder al menú lateral, teniendo acceso a las opciones disponibles
Precondiciones	<ul style="list-style-type: none"> • El usuario debe haber entrado a la aplicación • La ventana en la que se encuentre el usuario debe tener el icono de “menú lateral”
Postcondiciones	El usuario tendrá desplegado el menú lateral

*Tabla 27: Desplegar el menú lateral CU-16***ID CU-17**

Caso de uso	Acceder a Mi Perfil
Actor	Usuario registrado
Objetivo	El usuario podrá acceder a la sección Mi Perfil, para ver los datos de su perfil
Precondiciones	<ul style="list-style-type: none"> • El usuario debe estar registrado y logueado en la aplicación • El usuario debe tener desplegado el menú lateral
Postcondiciones	El usuario estará en la sección Mi Perfil

*Tabla 28: Acceder a Mi Perfil CU-17***ID CU-18**

Caso de uso	Editar perfil
Actor	Usuario registrado
Objetivo	El usuario podrá acceder a la ventana Configuración para editar su perfil desde la sección Mi Perfil
Precondiciones	<ul style="list-style-type: none"> • El usuario debe estar registrado y logueado en la aplicación • El usuario debe estar en la sección Mi Perfil
Postcondiciones	El usuario tendrá acceso a editar los datos de su perfil

Tabla 29: Editar perfil CU-18

ID CU-19

Caso de uso	Cerrar sesión
Actor	Usuario registrado
Objetivo	El usuario podrá cerrar la sesión actual
Precondiciones	<ul style="list-style-type: none"> • El usuario debe haberse registrado y logueado en la aplicación • El usuario debe estar en una de estas situaciones: <ul style="list-style-type: none"> a) Tener desplegado el menú lateral b) Estar en la sección Mi Perfil
Postcondiciones	El usuario ha cerrado su sesión

*Tabla 30: Cerrar sesión CU-19***ID CU-20**

Caso de uso	Contactar con CESyA
Actor	<ul style="list-style-type: none"> • Usuario registrado • Usuario invitado
Objetivo	El usuario podrá acceder a la sección “Contacta”, para contactar con CESyA, la organización fundadora de ACA
Precondiciones	<ul style="list-style-type: none"> • El usuario debe haber entrado en la aplicación • El usuario debe tener desplegado el menú lateral
Postcondiciones	El usuario tendrá acceso directo para contactar con CESyA

*Tabla 31: Contactar con CESyA CU-20***ID CU-21**

Caso de uso	Entrar en sección “Acerca de”
Actor	Usuario registrado
Objetivo	El usuario podrá seleccionar la sección “Acerca de”, para acceder a la ventana CESyA

Precondiciones	<ul style="list-style-type: none"> El usuario debe estar registrado y logueado en la aplicación El usuario debe tener desplegado el menú lateral
Postcondiciones	El usuario tendrá acceso directo a la ventana CESyA

Tabla 32: Entrar en sección "Acerca de..." CU-21

ID CU-22

Caso de uso	Salir del menú lateral
Actor	<ul style="list-style-type: none"> Usuario registrado Usuario no registrado
Objetivo	El usuario podrá salir del menú lateral
Precondiciones	<ul style="list-style-type: none"> El usuario debe haber entrado a la aplicación El usuario debe tener desplegado el menú lateral El usuario podrá hacer estas acciones para salir: <ul style="list-style-type: none"> a) Tocar icono "salir" b) Tocar en la parte oscura derecha de la pantalla c) Deslizar el menú lateral hacia la izquierda desde el borde derecho del menú
Postcondiciones	El usuario habrá salido del menú lateral

Tabla 33: Salir del menú lateral CU-22

5.2.3.3 Casos de uso de filtro de eventos

ID CU-23

Caso de uso	Filtrar evento por ciudad
Actor	Usuario registrado
Objetivo	El usuario podrá filtrar los eventos disponibles en una ciudad concreta
Precondiciones	<ul style="list-style-type: none"> El usuario debe haberse registrado y logueado en la aplicación El usuario debe estar en la pestaña "Eventos en..."
Postcondiciones	EL usuario tendrá acceso a los eventos de una ciudad concreta

Tabla 34: Filtrar evento por ciudad CU-23

ID CU-24

Caso de uso	Seleccionar una ciudad popular
Actor	Usuario registrado
Objetivo	El usuario podrá seleccionar, de forma directa, una de las ciudades más populares para aplicarlo como filtro
Precondiciones	<ul style="list-style-type: none"> El usuario debe haberse registrado y logueado en la aplicación El usuario debe estar en la pestaña “Eventos en...”
Postcondiciones	El usuario podrá filtrar por la ciudad que ha seleccionado

Tabla 35: Seleccionar una ciudad popular CU-24

ID CU-25

Caso de uso	Buscar una ciudad
Actor	Usuario registrado
Objetivo	El usuario podrá buscar una ciudad para filtrar los eventos a partir de la misma
Precondiciones	<ul style="list-style-type: none"> El usuario debe haberse registrado y logueado en la aplicación El usuario debe estar en la pestaña “Eventos en...” El usuario debe introducir una ciudad válida
Postcondiciones	EL usuario podrá seleccionar la ciudad buscada pulsando el botón “lupa”

Tabla 36: Buscar una ciudad CU-25

ID CU-26

Caso de uso	Seleccionar una ciudad
Actor	Usuario registrado
Objetivo	El usuario podrá seleccionar la ciudad que ha buscado

Precondiciones	<ul style="list-style-type: none"> • El usuario debe haberse registrado y logueado en la aplicación • El usuario debe estar en la pestaña “Eventos en...” • El usuario debe haber buscado la ciudad
Postcondiciones	El usuario podrá filtrar por la ciudad que ha seleccionado

Tabla 37: Seleccionar una ciudad CU-26

ID CU-27

Caso de uso	Filtrar evento por accesibilidad
Actor	Usuario registrado
Objetivo	El usuario podrá filtrar los eventos según los diferentes tipos de accesibilidad disponibles
Precondiciones	<ul style="list-style-type: none"> • El usuario debe haberse registrado y logueado en la aplicación • El usuario debe estar en la pestaña “Eventos en...” • El usuario solo podrá escoger un tipo de accesibilidad
Postcondiciones	EL usuario podrá filtrar por la accesibilidad que ha seleccionado

Tabla 38: Filtrar eventos por accesibilidad CU-27

ID CU-28

Caso de uso	Aplicar filtros
Actor	Usuario registrado
Objetivo	El usuario podrá aplicar la ciudad y la accesibilidad como filtro sobre los eventos disponibles
Precondiciones	<ul style="list-style-type: none"> • El usuario debe haberse registrado y logueado en la aplicación • El usuario debe estar en la pestaña “Eventos en...” • El usuario debe haber seleccionado una ciudad • El usuario debe haber seleccionado un tipo de accesibilidad
Postcondiciones	EL usuario podrá aplicar la ciudad y la accesibilidad que ha seleccionado y obtener los resultados del filtro

Tabla 39: Aplicar filtros CU-28

5.2.3.4 Casos de uso de almacenamiento de datos

ID CU-29

Caso de uso	Guardar cambios en Configuración
Actor	Usuario registrado
Objetivo	El usuario podrá guardar los cambios realizados en la ventana Configuración
Precondiciones	<ul style="list-style-type: none"> • El usuario debe estar registrado y logueado en la aplicación • El usuario debe estar en la ventana Configuración • El usuario debe haber editado algún campo de su perfil • Si el usuario desea cambiar su contraseña, debe repetirla en el campo “repite la password”
Postcondiciones	EL usuario tendrá cambiados los datos de su perfil

Tabla 40: Aplicar filtros CU-29

ID CU-30

Caso de uso	Guardar evento como favorito
Actor	Usuario registrado
Objetivo	El usuario podrá marcar un evento como favorito guardándolo en “Mi Archivo”, para poder consultarlo de forma más directa en cualquier momento
Precondiciones	<ul style="list-style-type: none"> • El usuario debe haberse registrado y logueado en la aplicación y estar en una categoría concreta • Debe haber eventos disponibles para esa categoría y para los filtros aplicados (por defecto o por el usuario)
Postcondiciones	EL usuario tendrá guardado los eventos que ha elegido como favoritos

Tabla 41: Guardar evento como favorito CU-30

ID CU-31

Caso de uso	Desmarcar evento de favorito
Actor	Usuario registrado
Objetivo	El usuario podrá eliminar un evento de “Mi Archivo”, quitándolo de la lista de favoritos

Precondiciones	<ul style="list-style-type: none"> El usuario debe haberse registrado y logueado en la aplicación El usuario debe haber marcado anteriormente el evento como favorito El usuario debe estar en una de estas situaciones: <ul style="list-style-type: none"> a) estar en la ventana “Mi Archivo” b) estar en el panel de eventos de una categoría
Postcondiciones	EL usuario ya no tendrá guardado el evento en “Mi Archivo”, pudiendo volver a marcarlo

Tabla 42: Desmarcar evento como favorito CU-31

5.3 Establecimiento de requisitos software

En este apartado se van a establecer los requisitos software que ha de seguir el sistema para conseguir un estado correcto del producto final.

Para la obtención de requisitos se ha partido del análisis de los casos de uso descrito anteriormente. A partir de los mismos se van a estudiar qué requisitos de software son necesarios para el desarrollo del sistema.

Se identifican tres tipos de requisitos software:

- **Requisitos funcionales:** conjunto de capacidades que cubren al sistema
- **Requisitos no funcionales:** conjunto de restricciones que debe seguir el sistema
- **Requisitos inversos:** conjunto de las acciones que el sistema no debe de hacer
- **Requisitos de interfaz:** conjunto de patrones de diseño que ha de seguir la interfaz de la aplicación

A continuación, se detalla la forma tabular que va a seguir la definición de cada requisito:

ID	XXX-NN
Nombre	
Descripción	
Necesidad	
Autor	
Fuente	
Dependencias	
Fecha de creación	
Fecha de modificación	

Tabla 43: Definición de requisitos software

Cada campo de la tabla sigue estas descripciones:

- **ID:** identificador alfanumérico del requisito. Se compone de dos partes:
 - XXX: siglas del tipo de requisito. El dominio de este campo es el siguiente:
 - RF: requisito funcional
 - RNF: requisito no funcional
 - RINV: requisito inverso
 - RINT: requisito de interfaz
 - NN: número del 00 al 99
- **Nombre:** nombre descriptivo del requisito
- **Descripción:** breve narrativa acerca de la funcionalidad del requisito, que exponga toda la información relevante del mismo
- **Necesidad:** indicador de la importancia del requisito en el sistema. El dominio de este campo es el siguiente:
 - Alta: requisito muy importante para el sistema
 - Media: requisito de importancia media para el sistema
 - Baja: requisito poco importante para el sistema
- **Autor:** autor del requisito. En este caso, el autor de todos los requisitos será el administrador. Este campo ayuda a mantener la trazabilidad del sistema.
- **Fuente:** origen del requisito, manteniendo la trazabilidad del sistema. El dominio de este campo es el siguiente:
 - CESyA: requisitos propios de la entidad origen colaboradora de este proyecto *(que haya 4 tipos de eventos...)*
 - UC3M: requisitos proporcionados por el director de este proyecto *(mas bien de mi tutor, que sea multiplataforma)*
 - Administrador del sistema: requisitos proporcionados por el administrador del sistema *(ideas que yo he tenido: login y registro...)*
- **Dependencias:** listado de otros productos de desarrollo de los cuales depende el requisito. Este campo mantiene la trazabilidad del sistema.
- **Fecha de creación:** fecha en la que dicho requisito se llevó a cabo
- **Fecha de modificación:** fecha en la que dicho requisito ha sido modificado. Este campo puede tener de 0 a N registros, ya que puede no haber sido cambiado, o haber sido modificado N veces.

5.3.1 Identificación de requisitos funcionales

ID	RF-00
Nombre	Entrar en la aplicación con <i>Android</i>
Descripción	El sistema debe permitir entrar en la aplicación con el sistema operativo Android
Necesidad	Alta
Autor	Administrador del sistema

Fuente	UC3M
Dependencias	
Fecha de creación	08-06-2018
Fecha de modificación	08-06-2018

Tabla 44: RF-00

ID RF-01

Nombre	Entrar en la aplicación con IOS
Descripción	El sistema debe permitir entrar en la aplicación con el sistema operativo IOS
Necesidad	Alta
Autor	Administrador del sistema
Fuente	UC3M
Dependencias	
Fecha de creación	06-08-2018
Fecha de modificación	06-08-2018

Tabla 45: RF-01

ID RF-02

Nombre	Registrarse en la aplicación
Descripción	El sistema debe permitir el registro en la aplicación
Necesidad	Alta
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	RF-00, RF-01
Fecha de creación	10-09-2018
Fecha de modificación	10-09-2018

Tabla 46: RF-02

ID RF-03

Nombre	Entrar sin registrarse en la aplicación
Descripción	El sistema debe permitir entrar en la aplicación sin registrarse, en modo invitado
Necesidad	Alta
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	RF-00, RF-01
Fecha de creación	10-09-2018
Fecha de modificación	10-09-2018

Tabla 47: RF-03

ID RF-04

Nombre	Establecer campo de nombre y apellidos en el formulario de registro
Descripción	El sistema debe pedir nombre y apellidos completos del usuario para llevar a cabo el registro en la aplicación
Necesidad	Alta
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	RF-00, RF-01, RF-02
Fecha de creación	10-09-2018
Fecha de modificación	10-09-2018

Tabla 48: RF-04

ID RF-05

Nombre	Establecer campo de correo electrónico en el formulario de registro
Descripción	El sistema debe pedir una cuenta de correo electrónico del usuario para llevar a cabo el registro en la aplicación

Necesidad	Alta
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	RF-00, RF-01, RF-02
Fecha de creación	10-09-2018
Fecha de modificación	10-09-2018

Tabla 49: RF-05

ID RF-06

Nombre	Establecer campo de nombre de usuario en el formulario de registro
Descripción	El sistema debe exigir crear un nombre de usuario para llevar a cabo el registro en la aplicación. Este nombre de usuario no debe existir ya en el sistema
Necesidad	Alta
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	RF-00, RF-01, RF-02
Fecha de creación	10-09-2018
Fecha de modificación	10-09-2018

Tabla 50: RF-06

ID RF-07

Nombre	Exigir nombre de usuario único en el formulario de registro
Descripción	El sistema debe exigir que no haya otro nombre de usuario igual ya registrado
Necesidad	Alta
Autor	Administrador del sistema
Fuente	Administrador del sistema

Dependencias	RF-00, RF-01, RF-02, RF-06
Fecha de creación	10-09-2018
Fecha de modificación	10-09-2018

Tabla 51: RF-07

ID RF-08

Nombre	Establecer campo de contraseña en el formulario de registro
Descripción	El sistema debe exigir crear una contraseña para llevar a cabo el registro en la aplicación.
Necesidad	Alta
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	RF-00, RF-01, RF-02
Fecha de creación	10-09-2018
Fecha de modificación	10-09-2018

Tabla 52: RF-08

ID RF-09

Nombre	Contraseña alfanumérica del registro
Descripción	El sistema debe exigir que la contraseña de la cuenta de usuario sea alfanumérica, formada tanto por números como por letras
Necesidad	Alta
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	RF-00, RF-01, RF-02, RF-08
Fecha de creación	10-09-2018
Fecha de modificación	10-09-2018

Tabla 53: RF-09

ID RF-10

Nombre	Repetir contraseña
Descripción	El sistema debe pedir la repetición de la contraseña para validar que el usuario la recuerda
Necesidad	Alta
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	RF-00, RF-01, RF-02, RF-08, RF-09
Fecha de creación	10-09-2018
Fecha de modificación	10-09-2018

Tabla 54: RF-10

ID RF-11

Nombre	Establecer campo de imagen en el formulario de registro
Descripción	El sistema debe incluir la selección de una imagen de perfil para completar el registro
Necesidad	Baja
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	RF-00, RF-01, RF-02
Fecha de creación	10-09-2018
Fecha de modificación	10-09-2018

Tabla 55: RF-11

ID RF-12

Nombre	Definir campo de imagen como opcional en el formulario de registro
Descripción	La imagen que exige el formulario de registro ha de ser opcional
Necesidad	Baja
Autor	Administrador del sistema

Fuente	Administrador del sistema
Dependencias	RF-00, RF-01, RF-02, RF-10
Fecha de creación	10-09-2018
Fecha de modificación	10-09-2018

Tabla 56: RF-12

ID RF-13

Nombre	Iniciar sesión en la aplicación
Descripción	El sistema debe permitir iniciar sesión para acceder a la aplicación
Necesidad	Alta
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	RF-00, RF-01, RF-02
Fecha de creación	10-09-2018
Fecha de modificación	10-09-2018

Tabla 57: RF-13

ID RF-14

Nombre	Iniciar sesión mediante nombre de usuario
Descripción	El sistema debe permitir iniciar sesión con el nombre de usuario de la cuenta registrada
Necesidad	Alta
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	RF-00, RF-01, RF-02, RF-12
Fecha de creación	10-09-2018
Fecha de modificación	10-09-2018

Tabla 58: RF-14

ID RF-15

Nombre	Modificar datos personales desde el perfil del usuario
Descripción	El sistema debe permitir al usuario modificar los datos de su cuenta desde la ventana del perfil de usuario
Necesidad	Media
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	RF-00, RF-01, RF-02, RF-12
Fecha de creación	12-09-2018
Fecha de modificación	12-09-2018

Tabla 59: RF-15

ID RF-16

Nombre	Modificar datos personales desde Configuración
Descripción	El sistema debe permitir al usuario modificar los datos de su cuenta desde la pestaña Configuración, ubicada en cualquier ventana de la aplicación
Necesidad	Media
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	RF-00, RF-01, RF-02, RF-12
Fecha de creación	12-09-2018
Fecha de modificación	12-09-2018

Tabla 60: RF-16

ID RF-17

Nombre	Definir género Museo
Descripción	El sistema debe incluir el género "Museo" entre los tipos de categorías
Necesidad	Alta

Autor	Administrador del sistema
Fuente	CESyA
Dependencias	RF-00, RF-01, RF-03, RF-02, RF-12
Fecha de creación	08-06-2018
Fecha de modificación	08-06-2018

Tabla 61: RF-17

ID RF-18

Nombre	Definir género Teatro
Descripción	El sistema debe incluir el género "Teatro" entre los tipos de categorías
Necesidad	Alta
Autor	Administrador del sistema
Fuente	CESyA
Dependencias	RF-00, RF-01, RF-03, RF-02, RF-12
Fecha de creación	08-06-2018
Fecha de modificación	08-06-2018

Tabla 62: RF-18

ID RF-19

Nombre	Definir género Cine
Descripción	El sistema debe incluir el género "Cine" entre los tipos de categorías
Necesidad	Alta
Autor	Administrador del sistema
Fuente	CESyA
Dependencias	RF-00, RF-01, RF-03, RF-02, RF-12
Fecha de creación	08-06-2018
Fecha de modificación	08-06-2018

Tabla 63: RF-19

ID RF-20

Nombre	Definir género Otros Eventos
Descripción	El sistema debe incluir el género “Otros eventos” entre los tipos de categorías, que incluirá eventos sin categorizar
Necesidad	Alta
Autor	Administrador del sistema
Fuente	CESyA
Dependencias	RF-00, RF-01, RF-03, RF-02, RF-12
Fecha de creación	08-06-2018
Fecha de modificación	08-06-2018

Tabla 64: RF-20

ID RF-21

Nombre	Proporcionar información sobre CESyA
Descripción	El sistema debe proporcionar al usuario información sobre CESyA, la organización que gestiona los eventos de ACA
Necesidad	Media
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	RF-00, RF-01, RF-03, RF-02, RF-12
Fecha de creación	14-09-2018
Fecha de modificación	14-09-2018

Tabla 65: RF-21

ID RF-22

Nombre	Contactar con CESyA mediante redes sociales
Descripción	El sistema debe permitir poner en contacto al usuario con CESyA mediante sus redes sociales
Necesidad	Media

Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	RF-00, RF-01, RF-03, RF-02, RF-12
Fecha de creación	14-09-2018
Fecha de modificación	14-09-2018

Tabla 66: RF-22

ID RF-23

Nombre	Acceder a eventos de una categoría
Descripción	El sistema debe permitir al usuario acceder a los eventos disponibles de cada categoría
Necesidad	Alta
Autor	Administrador del sistema
Fuente	CESyA
Dependencias	RF-00, RF-01, RF-03, RF-02, RF-12
Fecha de creación	14-09-2018
Fecha de modificación	14-09-2018

Tabla 67: RF-23

ID RF-24

Nombre	Mostrar información de un evento
Descripción	El sistema debe mostrar al usuario la información de un evento de una categoría concreta
Necesidad	Alta
Autor	Administrador del sistema
Fuente	CESyA

Dependencias	RF-00, RF-01, RF-03, RF-02, RF-12
Fecha de creación	14-09-2018
Fecha de modificación	14-09-2018

Tabla 68: RF-24

ID RF-25

Nombre	Incluir en la información de un evento sus servicios accesibles
Descripción	El sistema debe mostrar en la información de un evento los diferentes servicios de accesibilidad que comprende
Necesidad	Alta
Autor	Administrador del sistema
Fuente	CESyA
Dependencias	RF-00, RF-01, RF-03, RF-02, RF-12
Fecha de creación	14-09-2018
Fecha de modificación	14-09-2018

Tabla 69: RF-25

ID RF-26

Nombre	Incluir en la información de un evento su ubicación
Descripción	El sistema debe mostrar en la información de un evento la ubicación del lugar donde se desempeña
Necesidad	Alta
Autor	Administrador del sistema
Fuente	CESyA
Dependencias	RF-00, RF-01, RF-03, RF-02, RF-12
Fecha de creación	14-09-2018
Fecha de modificación	14-09-2018

Tabla 70: RF-26

ID RF-27

Nombre	Incluir en la información de un evento enlace a su página web
Descripción	El sistema debe mostrar en la información de un evento enlaces a sus páginas web para aportar información adicional
Necesidad	Alta
Autor	Administrador del sistema
Fuente	CESyA
Dependencias	RF-00, RF-01, RF-03, RF-02, RF-12
Fecha de creación	14-09-2018
Fecha de modificación	14-09-2018

Tabla 71: RF-27

ID RF-28

Nombre	Marcar evento como favorito
Descripción	El sistema debe permitir al usuario marcar un evento como favorito
Necesidad	Media
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	RF-00, RF-01, RF-02, RF-12, RF-23, RF-24
Fecha de creación	20-09-2018
Fecha de modificación	20-09-2018

Tabla 72: RF-28

ID RF-29

Nombre	Desmarcar evento como favorito
Descripción	El sistema debe permitir al usuario desmarcar un evento que se había guardado como favorito
Necesidad	Alta

Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	RF-00, RF-01, RF-02, RF-12, RF-23, RF-24, RF-28
Fecha de creación	20-09-2018
Fecha de modificación	20-09-2018

Tabla 73: RF-29

ID RF-30

Nombre	Filtrar eventos por provincia
Descripción	El sistema debe permitir al usuario filtrar los eventos por una provincia concreta
Necesidad	Alta
Autor	Administrador del sistema
Fuente	CESyA
Dependencias	RF-00, RF-01, RF-02, RF-12, RF-23, RF-24
Fecha de creación	14-09-2018
Fecha de modificación	14-09-2018

Tabla 74: RF-30

ID RF-31

Nombre	Filtrar eventos por servicio accesible
Descripción	El sistema debe permitir al usuario filtrar los eventos por servicio de accesibilidad concreto
Necesidad	Alta
Autor	Administrador del sistema
Fuente	CESyA
Dependencias	RF-00, RF-01, RF-02, RF-12, RF-23, RF-24, RF-30
Fecha de creación	14-09-2018
Fecha de modificación	14-09-2018

Tabla 75: RF-31

ID RF-32

Nombre	Mostrar acceso directo a provincias populares
Descripción	El sistema debe mostrar un acceso directo a las provincias más populares que componen los sistemas de CESyA
Necesidad	Media
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	RF-00, RF-01, RF-02, RF-12, RF-23, RF-24, RF-30
Fecha de creación	15-09-2018
Fecha de modificación	15-09-2018

Tabla 76: RF-32

ID RF-33

Nombre	Buscar provincia
Descripción	El sistema debe permitir al usuario buscar la provincia por la que desea filtrar los eventos
Necesidad	Alta
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	RF-00, RF-01, RF-02, RF-12, RF-23, RF-24, RF-30
Fecha de creación	15-09-2018
Fecha de modificación	15-09-2018

Tabla 77: RF-33

ID RF-34

Nombre	Filtrar eventos con los dos filtros
Descripción	El sistema debe permitir al usuario filtrar los eventos siempre que se hayan seleccionado ambos filtros (provincia y servicio accesible)

Necesidad	Alta
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	RF-00, RF-01, RF-02, RF-12, RF-23, RF-24, RF-30, RF-31
Fecha de creación	15-09-2018
Fecha de modificación	15-09-2018

Tabla 78: RF-34

ID RF-35

Nombre	Cerrar sesión
Descripción	El sistema debe permitir al usuario cerrar su sesión
Necesidad	Baja
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	RF-00, RF-01, RF-02, RF-12
Fecha de creación	10-09-2018
Fecha de modificación	10-09-2018

Tabla 79: RF-35

ID RF-36

Nombre	Permitir a usuario registrado entrar como invitado
Descripción	El sistema puede permitir el acceso en modo invitado a un usuario con cuenta en ACA, siempre que haya cerrado su sesión
Necesidad	Baja
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	RF-00, RF-01, RF-02

Fecha de creación	10-09-2018
Fecha de modificación	10-09-2018

Tabla 80: RF-36

ID RF-37

Nombre	Mostrar eventos en modo invitado
Descripción	El sistema debe mostrar al usuario invitado los eventos accesibles en Madrid con accesibilidad completa
Necesidad	Alta
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	RF-00, RF-01
Fecha de creación	10-09-2018
Fecha de modificación	10-09-2018

Tabla 81: RF-37

5.3.2 Identificación de requisitos no funcionales

ID RNF-00

Nombre	Aplicación multiplataforma
Descripción	El sistema debe estar orientado a la multiplataforma, alejado de los desarrollos nativos
Necesidad	Alta
Autor	Administrador del sistema
Fuente	UC3M
Dependencias	
Fecha de creación	08-06-2018
Fecha de modificación	08-06-2018

Tabla 82: RNF-00

ID RNF-01

Nombre	Plataforma <i>Android</i>
Descripción	El sistema se debe desplegar en la plataforma <i>Android</i>
Necesidad	Alta
Autor	Administrador del sistema
Fuente	UC3M
Dependencias	RNF-00
Fecha de creación	08-06-2018
Fecha de modificación	08-06-2018

Tabla 83: RNF-01

ID RNF-02

Nombre	Plataforma <i>IOS</i>
Descripción	El sistema se debe desplegar en la plataforma <i>IOS</i>
Necesidad	Alta
Autor	Administrador del sistema
Fuente	UC3M
Dependencias	RNF-00
Fecha de creación	08-06-2018
Fecha de modificación	08-06-2018

Tabla 84: RNF-02

ID RNF-03

Nombre	Conexión a internet
Descripción	El sistema requiere de conexión a internet para su despliegue en cualquier plataforma
Necesidad	Alta

Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	
Fecha de creación	10-09-2018
Fecha de modificación	10-09-2018

Tabla 85: RNF-03

ID RNF-04

Nombre	Desarrollo y compilación del sistema en Apache Córdoba
Descripción	El sistema se debe desarrollar y compilar mediante el entorno de desarrollo Apache Córdoba
Necesidad	Alta
Autor	Administrador del sistema
Fuente	UC3M
Dependencias	RF-00
Fecha de creación	08-06-2018
Fecha de modificación	08-06-2018

Tabla 86: RNF-04

ID RNF-05

Nombre	Desarrollo en HTML
Descripción	La aplicación se debe programar en el lenguaje de marcas HTML
Necesidad	Alta
Autor	Administrador del sistema
Fuente	UC3M
Dependencias	RNF-04
Fecha de creación	08-06-2018

Fecha de modificación	08-06-2018
-----------------------	------------

Tabla 87: RNF-05

ID	RNF-06
Nombre	Desarrollo en JavaScript
Descripción	La aplicación se debe programar en el lenguaje de orientado a prototipos JavaScript
Necesidad	Alta
Autor	Administrador del sistema
Fuente	UC3M
Dependencias	RNF-04
Fecha de creación	08-06-2018
Fecha de modificación	08-06-2018

Tabla 88: RNF-06

ID	RF-07
Nombre	Desarrollo en CSS
Descripción	La aplicación se debe programar en el lenguaje CSS para proporcionar el estilo a la interfaz
Necesidad	Alta
Autor	Administrador del sistema
Fuente	UC3M
Dependencias	RNF-04
Fecha de creación	08-06-2018
Fecha de modificación	08-06-2018

Tabla 89: RNF-07

ID RNF-08

Nombre	Base de datos en tiempo real
Descripción	El sistema requiere de una base de datos en tiempo real para almacenar los datos de cada cliente
Necesidad	Alta
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	
Fecha de creación	18-09-2018
Fecha de modificación	18-09-2018

Tabla 90: RNF-08

ID RNF-09

Nombre	Almacenar y extraer información de los usuarios de la base de datos del sistema
Descripción	La información de los usuarios ha de almacenarse en la base de datos del sistema
Necesidad	Alta
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	RNF-08
Fecha de creación	18-09-2018
Fecha de modificación	18-09-2018

Tabla 91: RNF-09

ID RNF-10

Nombre	Extraer información de los eventos de servicios web de CESyA
Descripción	La información de los eventos de la aplicación han de extraerse de los <i>Web Services</i> de CESyA

Necesidad	Alta
Autor	Administrador del sistema
Fuente	UC3M
Dependencias	
Fecha de creación	08-06-2018
Fecha de modificación	08-06-2018

Tabla 92: RNF-10

ID RNF-11

Nombre	Peticiones mediante método POST a los servicios web de CESyA
Descripción	Las peticiones que se realicen a los servicios web deben ser mediante el método POST
Necesidad	Alta
Autor	Administrador del sistema
Fuente	UC3M
Dependencias	RNF-10
Fecha de creación	08-06-2018
Fecha de modificación	08-06-2018

Tabla 93: RNF-11

5.3.3 Identificación de requisitos inversos

ID RINV-00

Nombre	Filtrar eventos en modo invitado
Descripción	El sistema no debe permitir que un usuario invitado filtre eventos
Necesidad	Alta
Autor	Administrador del sistema
Fuente	Administrador del sistema

Dependencias	
Fecha de creación	20-09-2018
Fecha de modificación	20-09-2018

Tabla 94: RINV-00

ID RINV -01

Nombre	Tener perfil en modo invitado
Descripción	El sistema no debe permitir que un usuario invitado tenga un perfil con sus datos personales
Necesidad	Alta
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	
Fecha de creación	20-09-2018
Fecha de modificación	20-09-2018

Tabla 95: RINV-01

ID RINV -02

Nombre	Filtrar eventos con una sola opción
Descripción	El sistema no debe permitir que un usuario registrado aplique solo una opción en el filtro de eventos
Necesidad	Alta
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	
Fecha de creación	18-09-2018
Fecha de modificación	18-09-2018

Tabla 96: RINV-02

ID RINV -03

Nombre	Permitir a usuario invitado entrar como registrado
Descripción	El sistema no debe permitir que un usuario invitado entre a la aplicación como usuario registrado si no tiene cuenta en ACA
Necesidad	Alta
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	
Fecha de creación	20-09-2018
Fecha de modificación	20-09-2018

Tabla 97: RINV-03

5.3.4 Identificación de requisitos de interfaz

ID RINT-00

Nombre	Establecer colores de la aplicación
Descripción	En los colores del sistema debe predominar los tonos azules claros y oscuros. Para los contenedores de eventos se acompañará de tonos verdes
Necesidad	Media
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	
Fecha de creación	21-09-2018
Fecha de modificación	21-09-2018

Tabla 98: RINT-00

ID RINT-01

Nombre	Acceder al catálogo de categorías desde pantalla principal
Descripción	El sistema debe permitir al usuario acceder a las diferentes categorías desde la ventana principal

Necesidad	Alta
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	
Fecha de creación	21-09-2018
Fecha de modificación	21-09-2018

Tabla 99: RINT-01

ID RINT-02

Nombre	Acceder al almacén de eventos desde pantalla principal
Descripción	El sistema debe permitir al usuario acceder al listado de los eventos que ha guardado desde la pantalla principal
Necesidad	Alta
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	
Fecha de creación	21-09-2018
Fecha de modificación	21-09-2018

Tabla 100: RINT-02

ID RINT -03

Nombre	Definir disposición de las categorías
Descripción	Las cuatro categorías han de mostrarse en rectángulos, uno para cada género, ocupando más de la mitad de la pantalla principal
Necesidad	Baja
Autor	Administrador del sistema
Fuente	Administrador del sistema

Dependencias	
Fecha de creación	21-09-2018
Fecha de modificación	21-09-2018

Tabla 101: RINT-03

ID RINT-04

Nombre	Definir disposición de los eventos
Descripción	Los eventos de cada categoría deben estar cada uno en un contenedor, de colores variados, siguiendo diferentes tamaños
Necesidad	Baja
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	
Fecha de creación	21-09-2018
Fecha de modificación	21-09-2018

Tabla 102: RINT-04

ID RINT-05

Nombre	Incluir imagen descriptiva en cada categoría
Descripción	Los contenedores de cada categoría deben llevar una imagen que acompañe al título del genero
Necesidad	Alta
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	
Fecha de creación	21-09-2018
Fecha de modificación	21-09-2018

Tabla 103: RINT-05

ID RINT-06

Nombre	Incluir imagen en cada ciudad
Descripción	Los contenedores de cada ciudad deben llevar una imagen representativa de la provincia
Necesidad	Media
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	
Fecha de creación	22-09-2018
Fecha de modificación	22-09-2018

Tabla 104: RINT-06

ID RINT-07

Nombre	Establecer icono de evento favorito
Descripción	El icono que indicará que un evento puede ser favorito tendrá borde blanco y no estará relleno. Cuando el evento sea favorito, este icono estará relleno de blanco
Necesidad	Media
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	
Fecha de creación	21-09-2018
Fecha de modificación	21-09-2018

Tabla 105: RINT-07

ID RINT-08

Nombre	Establecer color de los iconos
Descripción	Los iconos de accesos a pestañas deben encajar con las tonalidades de la aplicación, siendo azules oscuros

Necesidad	Media
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	
Fecha de creación	21-09-2018
Fecha de modificación	21-09-2018

Tabla 106: RINT-08

ID RINT-09

Nombre	Definir fuente del texto de la aplicación
Descripción	Para algún texto concreto se usará fuentes de Google. Para el resto, se seguirá la familia <i>Arial</i>
Necesidad	Media
Autor	Administrador del sistema
Fuente	Administrador del sistema
Dependencias	
Fecha de creación	21-09-2018
Fecha de modificación	21-09-2018

Tabla 107: RINT-09

5.4 Identificación de subsistemas de análisis

En este apartado se definen los subsistemas que componen al sistema del proyecto. Se pueden distinguir cuatro subsistemas:

- Acceso a la aplicación
- Navegación por la aplicación
- Filtro de eventos
- Almacenamiento de datos

Estos subsistemas siguen el patrón MVC (Modelo, Vista y Controlador), siendo:

- **Modelo:** subsistema de almacenamiento de datos
- **Vista:** subsistema de navegación por la aplicación
- **Controlador:** subsistemas de acceso a la aplicación y filtro de eventos

La estructura del sistema, siguiendo el patrón MVC, sería la siguiente:

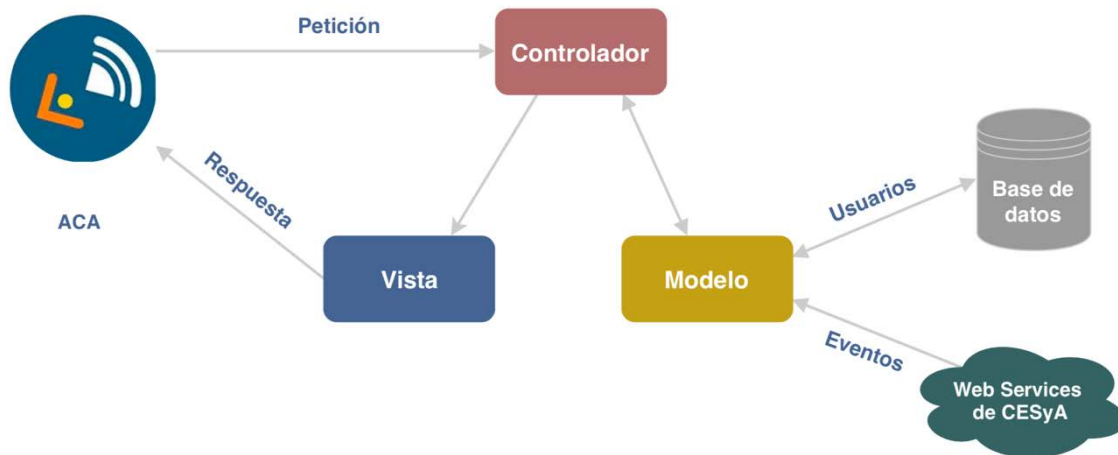


Ilustración 13: Patrón MVC del sistema

5.4.1 Definición de cada subsistema

5.4.1.1 Acceso a la aplicación

Este subsistema gestiona los accesos a la aplicación. Como se ha definido anteriormente, el usuario puede acceder al sistema habiéndose registrado o en modo invitado.

Este componente será el encargado de actuar como *Controlador* para administrar el acceso del usuario al sistema.

5.4.1.2 Navegación por la aplicación

Este subsistema trata la navegación del usuario por la aplicación. Cuenta con la *Vista* del sistema, ya que es el encargado de mostrar al usuario las respuestas a sus peticiones.

5.4.1.3 Filtro de eventos

Este subsistema gestiona el filtro que aplica el usuario a los eventos. Forma parte del *Controlador* del sistema, ya que ha de gestionar las peticiones enlazadas con los datos que se obtienen del *Web Services*.

5.4.1.4 Almacenamiento de datos

Este subsistema gestiona el *Modelo* del sistema. Comprende tanto el almacenamiento de la información de usuarios en la base de datos, como la lectura de ésta y del *Web Services*, que contiene la información de los eventos. Esta extracción y carga de la información se especifica con más detalle en la definición del modelo de datos.

Proporciona la persistencia de los datos del sistema.

5.4.2 Relaciones entre subsistemas

Para asegurar la consistencia del sistema, se muestra las relaciones entre cada uno de los subsistemas:

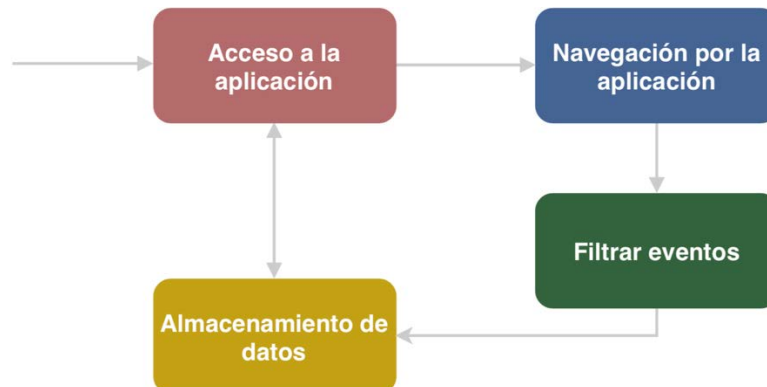


Ilustración 14: Relación entre los subsistemas

5.5 Elaboración del modelo de datos

Este proyecto sigue una arquitectura distribuida. Esto significa que los datos que proveen a la aplicación provienen de diferentes servicios. La información que gestiona el sistema se divide en dos secciones:

1. Datos referentes a los **usuarios**, como su información personal o los eventos que ha marcado como favoritos.
2. Datos referentes a los **eventos**, como la información propia de cada evento, el tipo de accesibilidad, el género al que pertenece y la cantidad de eventos para una determinada ciudad.

Cada una de estas secciones provienen de los siguientes servicios:

Base de datos

El sistema cuenta con una base de datos en tiempo real, proporcionada por Realtime Database de Firebase (Google) (17). Se trata de una base de datos no relacional.

Los usuarios se almacenan, se leen y se gestionan dentro de esta base de datos. Cuenta con una única colección, *Users*, donde está toda la información.

Para almacenar los eventos que el usuario desea guardar, obtiene su identificador del *web services*. Será el *Controlador* el encargado de dar esta información a la base de datos del propio sistema.

Web Services de CESyA

Toda la información de los eventos la proporcionan servicios web de la organización CESyA. Es por ello por lo que solo es necesaria la lectura de esta información, y se realiza sobre estos servicios. El *Controlador* se encarga de traspasar estos datos a la *Vista* y al *Modelo*.

Este proyecto no tiene conocimiento del modelo de datos que siguen estos servicios web. Por este motivo se va a analizar de una forma abstracta y conceptual, para poder estudiar su relación con el sistema.

Analizadas ya tanto la extracción como la carga de la información del sistema, se puede definir el modelo de datos. Este modelo va a seguir la estructura física de la base de datos relacionada con la arquitectura conceptual de la información que se extrae de los *Web Services* (ya que no se conoce la totalidad de su modelo).

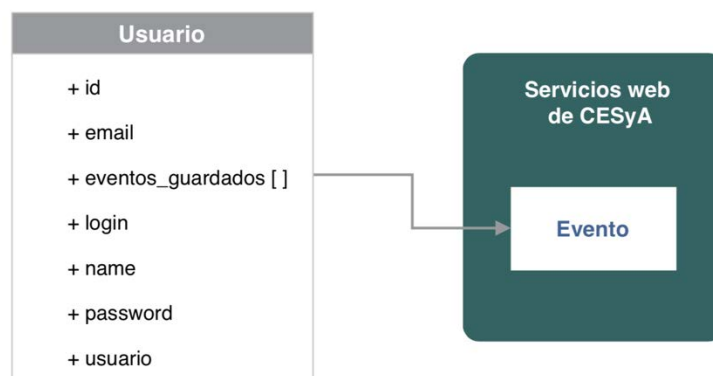


Ilustración 15: Modelo de datos del sistema

Como se ha explicado en este mismo apartado, el sistema consta únicamente de una colección en la base de datos. El resto de información que trata el sistema se obtiene de los *Web Services* de CESyA.

La colección *Users* contiene estos atributos:

- *Id*: campo alfanumérico que identifica de forma única a cada usuario
- *Email*: correo electrónico del usuario
- *Eventos_guardados*: array clave-valor de cada evento que el usuario ha guardado
- *Name*: nombre completo del usuario
- *Password*: contraseña del usuario
- *Usuario*: nombre de usuario

5.6 Definición de las interfaces de usuario

A continuación, se van a definir las interfaces de usuario que va a seguir la aplicación. Estas interfaces deben cumplir los siguientes principios de usabilidad, según las heurísticas de Nielsen (18):

- **Visibilidad del estado del sistema**: aportar al usuario una retroalimentación adecuada
- **Correspondencia entre el sistema y el mundo real**: la información que ofrezca la interfaz debe aparecer en un orden natural y lógico

- **Control de usuario y libertad:** la interfaz debe soportar acciones de deshacer, rehacer y salidas de emergencia
- **Prevención de errores:** evitar situaciones que puedan llegar a error al usuario
- **Coherencia y estándares:** las palabras, situaciones o acciones de la interfaz deben ser consistentes
- **Reconocimiento en vez de recordar:** el usuario debe saber en todo momento dónde se encuentra, haciendo visibles objetos, acciones y opciones
- **Flexibilidad y eficiencia:** la aplicación debe ser útil para todo tipo de usuario, adaptándose a las acciones más frecuentes
- **Diseño estético y minimalista:** las interfaces deben mostrar información relevante y necesaria
- **Ayuda y documentación:** la aplicación debe contener documentación y ayuda guía fácil de utilizar
- **Ayudar al usuario a reconocer, diagnosticar y recuperar errores:** la aplicación debe expresar mensajes de error en lenguaje sencillo, indicando el problema y sugerir una solución

Para trazar de una forma más sencilla las interfaces y los casos de uso, se han dividido según su funcionalidad, al igual que los casos de uso:

- Interfaces de usuario de acceso a la aplicación
- Interfaces de usuario de navegación por la aplicación
- Interfaces de usuario de filtro de eventos
- Interfaces de usuario de almacenamiento de datos

5.6.1 Interfaces de acceso a la aplicación

La primera interfaz de usuario muestra la pantalla inicial de la aplicación:

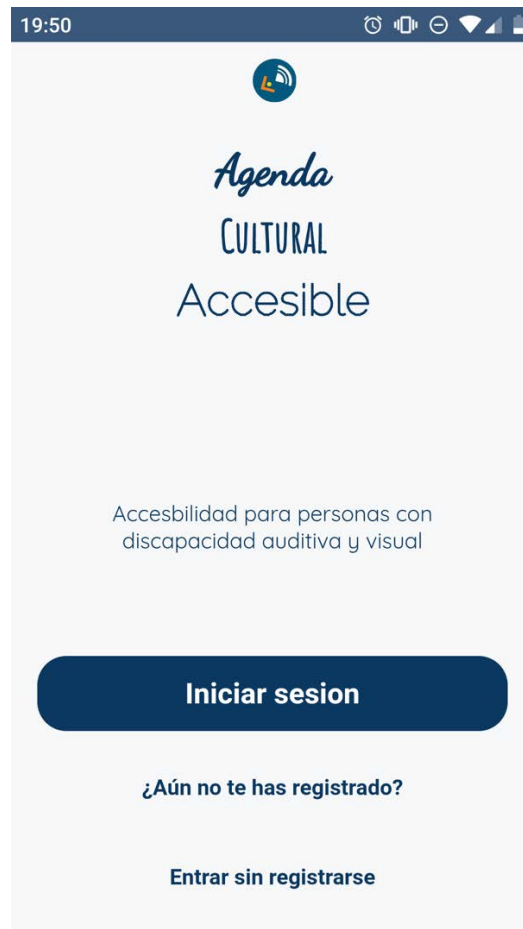


Ilustración 16: Pantalla inicial de la aplicación IU-00

En cuanto el usuario entre en la aplicación, se topará con esta pantalla. Muestra las diferentes opciones de acceso a la aplicación:

- **Iniciar sesión:** entrar a la aplicación con tu usuario y contraseña
- **¿Aún no te has registrado?:** registrarse en la aplicación creándose una cuenta de usuario para poder iniciar sesión
- **Entrar sin registrarse:** entrar a la aplicación en modo invitado, sin tener una cuenta de usuario en la aplicación

Estas opciones siguen un orden lógico. La primera vez que el usuario entre a la aplicación, lo normal es que se registre. Pero esa acción se realiza una única vez, y se espera que hará más repetidas veces el inicio de sesión. Por ello esta opción se encuentra la superior y la más marcada, para que el usuario sea lo primero que vea. La última opción es que entre sin registrarse, ya que es la alternativa más restringida de acceso al sistema.

Además, cuenta tanto con el logo como con el nombre de la aplicación para situar al usuario. Contiene una breve descripción del objetivo del sistema.

Para iniciar sesión se establece esta interfaz:


A mobile application login screen. At the top, a status bar shows the time 19:51 and various icons. Below it, a navigation bar contains a back arrow and a circular logo. The main heading is 'Accede'. There are two input fields: 'Usuario' with placeholder text 'nombre de usuario' and 'Password *' with placeholder text 'contraseña'. Below these is a blue button labeled '¡Entra!'. At the bottom, a note states '(*): campos obligatorios'.

Ilustración 17: Inicio de sesión IU-01

El formulario necesario para iniciar sesión se compone del nombre de usuario y contraseña que se escogió cuando se registró. En cuanto se pulse sobre el botón “¡Entra!” se accede a la pantalla principal de la aplicación. Además, se avisa al usuario de qué campos son obligatorios mediante un mensaje debajo del botón, ayudándole a evitar errores.

El botón en forma de flecha en la esquina izquierda superior indica la posibilidad de retornar a la ventana inicial. Esto facilita el control y la libertad del usuario sobre el sistema. El logo de la aplicación no tiene ninguna acción.

Realizando este formulario el usuario se registra en la aplicación:

19:51

<

Crea tu cuenta

Nombre y apellidos *

nombre y apellidos

Email *

email

Siguiente

(*) campos obligatorios

19:51

<

Crea tu cuenta

Usuario *

nombre de usuario

Password *

password

Imagen de perfil

Seleccionar archivo Ningún archivo seleccionado

¡Entra!

(*) campos obligatorios

Ilustración 18: Registro 1 IU-02

Ilustración 19: Registro 2 IU-03

Como se ha definido anteriormente tanto en los casos de uso como en los requisitos, los datos necesarios para el registro son:

- *Nombre y apellidos*: nombre completo del usuario
- *Email*: correo electrónico del usuario
- *Usuario*: nombre de usuario único que identifique al usuario
- *Password*: contraseña alfanumérica que escoja el usuario
- *Imagen de perfil*: imagen que describa al usuario, siempre que lo desee

Este formulario se ha dividido en dos pantallas para que sea más visual al usuario. En la primera, muestra únicamente la información a rellenar de carácter personal. El botón “*Siguiente*”

encadena la segunda parte del formulario, la cual muestra los campos a rellenar de información más administrativa para el sistema.

En ambas ventanas también se muestra mensajes de ayuda al usuario y la posibilidad del volver a la pantalla anterior.

La opción “Entrar sin registrarse” dirige directamente a la pantalla principal. Esta interfaz se va a mostrar en la siguiente sección, ya que pertenece a la navegación de la aplicación.

5.6.2 Interfaces de navegación por la aplicación

La pantalla principal de la aplicación es la siguiente:

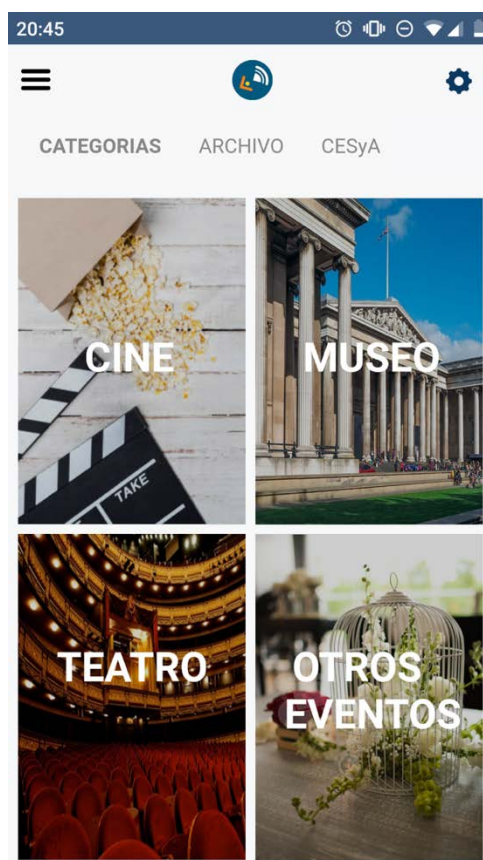


Ilustración 20: Pantalla principal IU-04

Esta pantalla es igual tanto para el usuario registrado como para el usuario invitado. Desde ella se accede a todas las secciones de la aplicación.

Por defecto, se abre en la pestaña “Categorías”, que contiene los cuatro tipos de eventos que ofrece el sistema: cine, museo, teatro y otros eventos. Pulsando sobre cada contenedor de las imágenes se accede a las pantallas de cada género.

La pestaña seleccionada de la barra horizontal se muestra remarcada para ayudar al situar al usuario. Dentro de esta misma pantalla, el resto de pestañas de dicha barra tienen estas interfaces:



Ilustración 21: Mi Archivo IU-05

Ilustración 22: CESyA IU-06

La ilustración 20 muestra Mi Archivo, el almacén de eventos guardados por el usuario. La ilustración Y muestra información sobre CESyA, la organización que gestiona los eventos accesibles.

Los botones superiores de esta pantalla se corresponden con la ventana “Configuración” (icono en la esquina superior derecha) y con un menú lateral (icono en la esquina superior izquierda):

La ventana de *Configuración* se detalla en las interfaces de almacenamiento de datos.

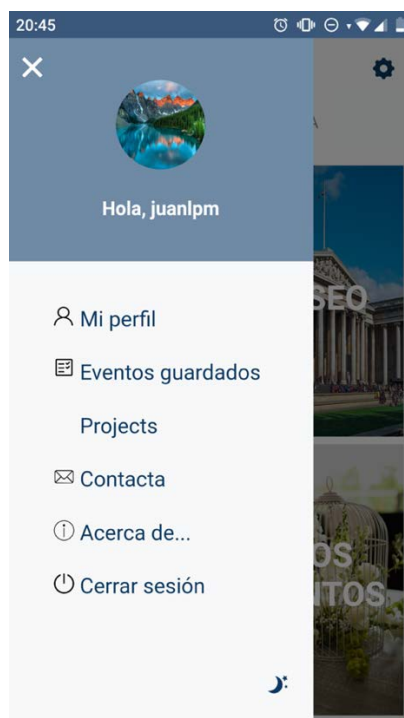


Ilustración 23: Menú lateral IU-07

Este menú está disponible en la mayoría de pantallas de la aplicación. Contiene accesos directos a las ventanas útiles para el usuario:

- Mi Perfil: acceder al perfil del usuario
- Eventos guardados: acceder a “Mi Archivo”
- Contacta
- Acerca de...
- Cerrar sesión: cerrar la sesión del usuario. Esta opción retorna a la pantalla inicial, borrando los datos de la sesión

Para salir de este menú bien se pulsa sobre el icono “X” de la esquina superior izquierda, o bien se desliza hacia la derecha el menú desde la zona sombreada de la pantalla.

La sección “Mi perfil” sigue esta interfaz:

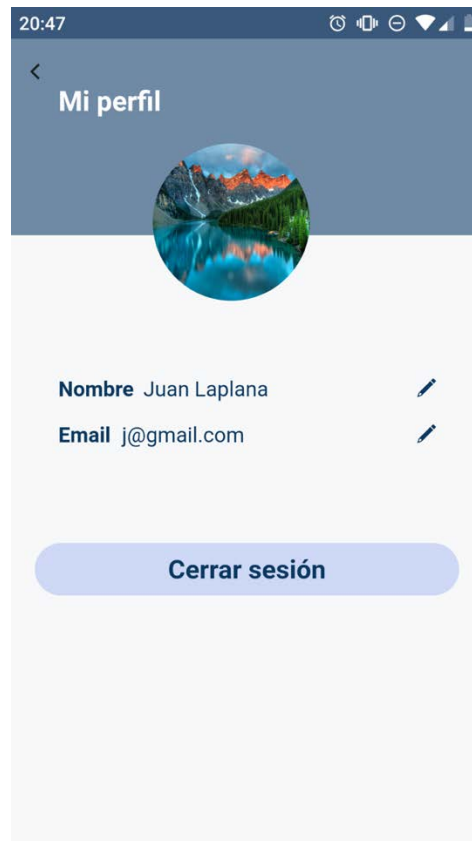


Ilustración 24: Mi Perfil IU-08

Es una pantalla sencilla, en la que se muestra la información del usuario. Da la posibilidad de cerrar sesión, volviendo así a la pantalla inicial con los datos de la sesión eliminados.

Acabando con la pantalla principal, esta es la interfaz de las pantallas de cada categoría. Como es igual para cada género (cambiando los eventos correspondientes), se va a mostrar únicamente para **museo**:



Ilustración 25: Eventos IU-09

Cada contenedor coloreado es uno de los eventos disponibles. Por defecto, se muestran los eventos con accesibilidad completa disponibles en Madrid. La pestaña “*Eventos en Madrid*” permite al usuario filtrar los eventos. Estas pantallas se verán en las interfaces de filtro de eventos.

Pulsando sobre cada evento se muestra su información:

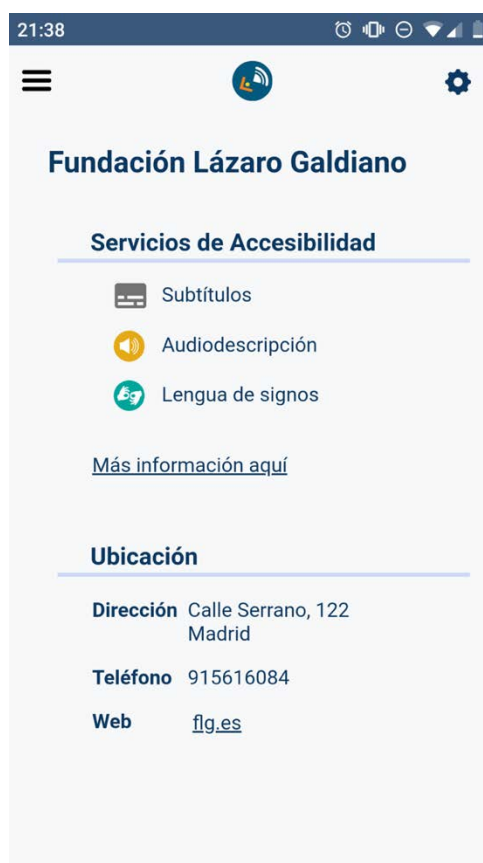


Ilustración 26: Información evento IU-10

En concreto, esta pantalla muestra:

- **Servicios de accesibilidad** que ofrece el evento. Cada uno del título de estos servicios viene acompañado de un icono representativo de la accesibilidad. Se muestra disponible también enlace externo para obtener más información.
- **Ubicación** del lugar donde se celebra dicho evento. Se compone de la dirección, el teléfono y un acceso directo a la web del evento.

5.6.3 Interfaces de filtro de eventos

Dentro de cada género, el usuario tiene la posibilidad de filtrar eventos por:

- **Ciudad** donde se realice el evento
- **Servicio de accesibilidad** que ofrezca el evento

Para filtrar los eventos, el usuario debe entrar en la sección “Eventos en Madrid”:

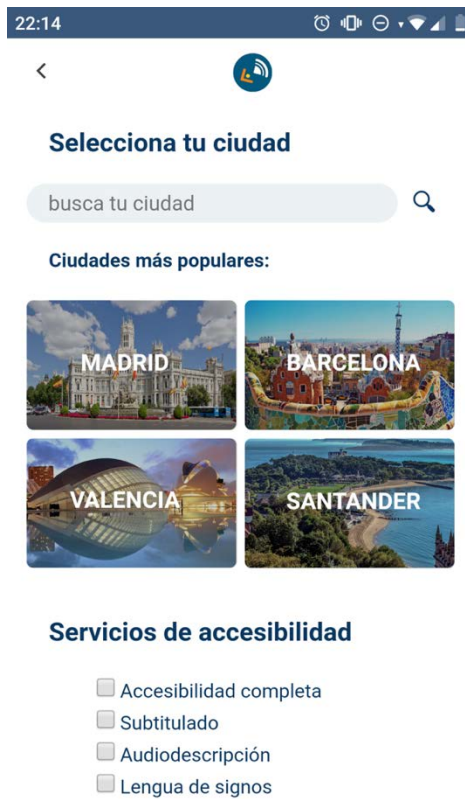


Ilustración 27: Filtrar evento IU-11

Para filtrar por ciudad, existen varias opciones:

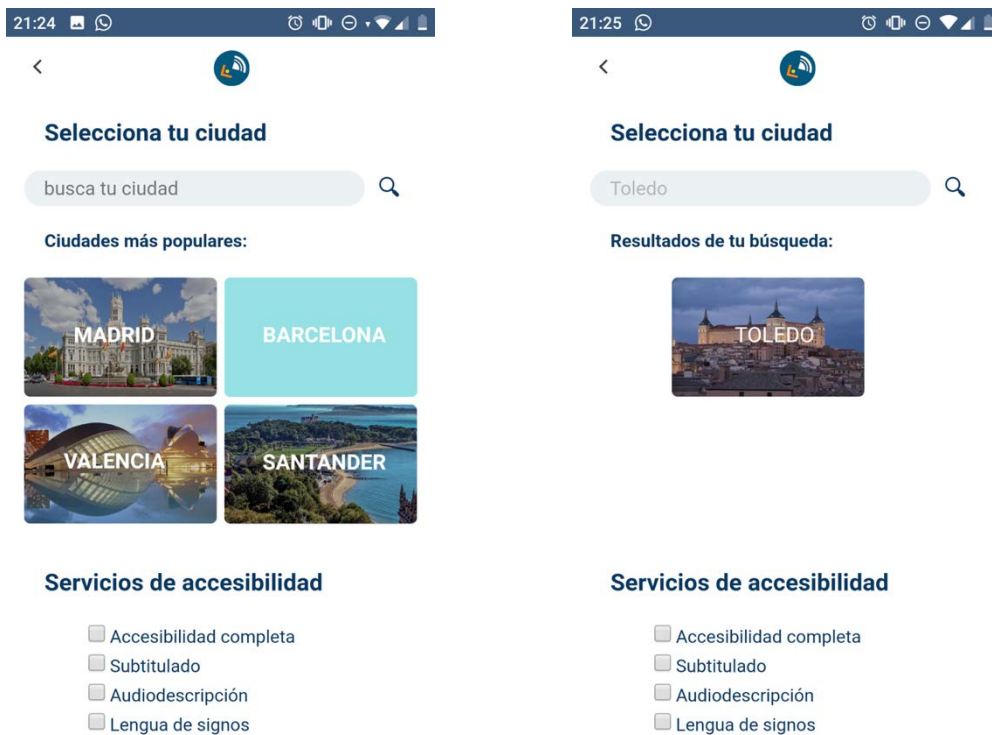


Ilustración 28: Seleccionar ciudad popular IU-12

Ilustración 29: Buscar ciudad IU-13

- La ilustración X muestra cómo al abrir la ventana se muestra por defecto cuatro de las ciudades más populares del sistema. Si la ciudad por la que desea filtrar el usuario se encuentra entre esas cuatro, puede seleccionarla directamente.
- En caso de que la ciudad que el usuario desea no se encuentra entre las populares, tiene la posibilidad de introducir su nombre en el campo superior y buscarlo con el icono de al lado en forma de lupa. Por ejemplo, si el usuario quiere filtrar por Toledo, la seleccionaría una vez encontrada, como muestra la ilustración Y.

El usuario debe escribir correctamente el nombre de la ciudad. En caso contrario, no habrá resultados a su búsqueda.

Una vez seleccionada la ciudad, es obligatorio seleccionar un tipo de accesibilidad para continuar. Si no se ha seleccionado alguno de los dos filtros, no se podrán aplicar:

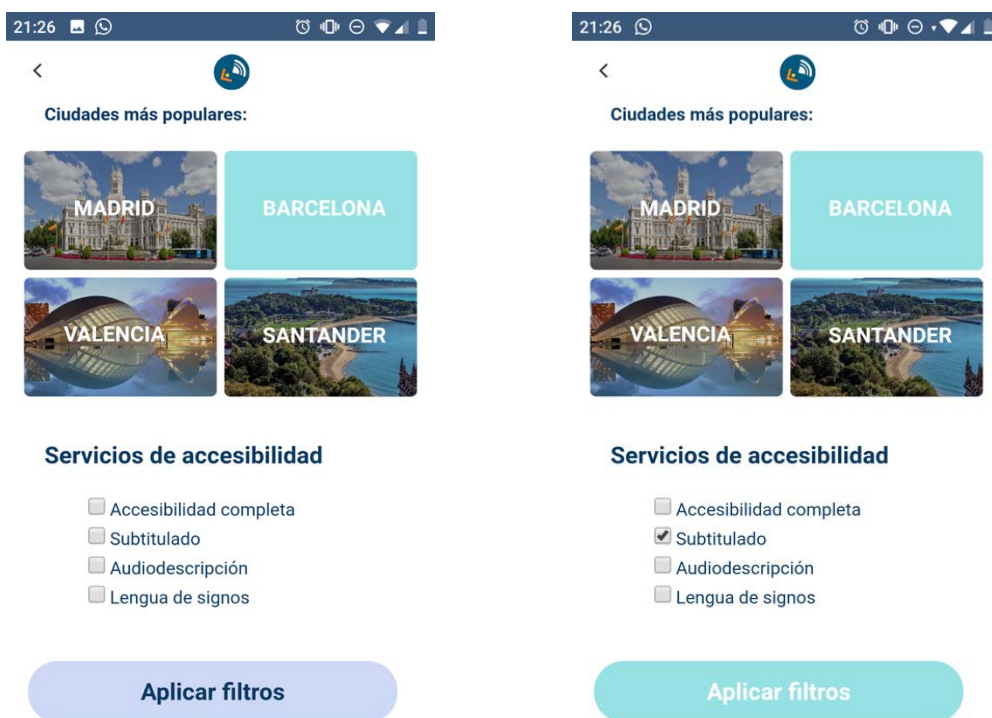


Ilustración 30: Seleccionar solo filtro ciudad IU-11

Ilustración 31: Seleccionar ciudad y accesibilidad IU-11

La ilustración 29 muestra cómo el botón “Aplicar filtros” no se ha activado, ya que no ha cambiado de color. La ilustración Y muestra, sin embargo, como tras seleccionarse también el servicio de accesibilidad se activa el botón. De esta manera se muestra al usuario la necesidad de escoger los dos filtros a la vez.

Si los filtros se han seleccionado correctamente, al darle al botón “Aplicar filtros” la aplicación carga a los eventos de dicha ciudad con el tipo de accesibilidad seleccionado:

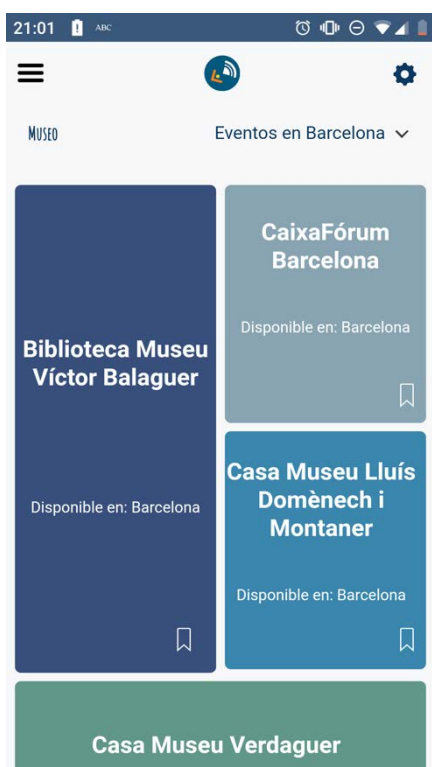


Ilustración 32: Filtros aplicados sobre eventos 1 IU-12

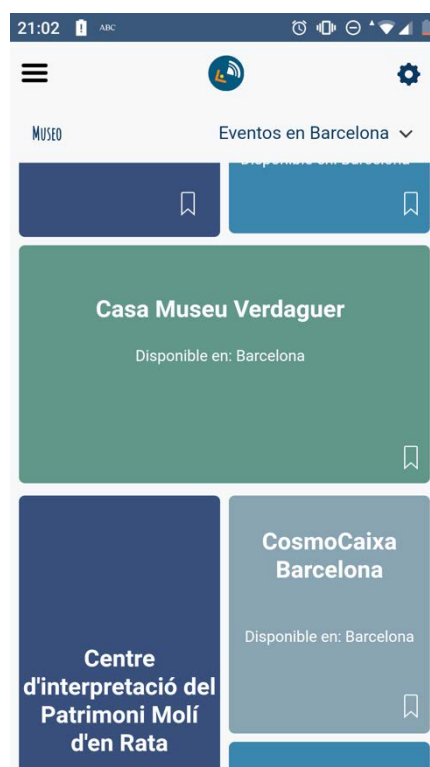


Ilustración 33: Filtros aplicados sobre eventos 2 IU-12

Como se aprecia en las ilustraciones anteriores, se muestran los eventos de Barcelona. Además, ha cambiado la pestaña “Eventos en ...” a la ciudad actual.

Es posible que el usuario busque una ciudad la cual no cuente con eventos accesibles del género en ese momento. La aplicación le avisará de ello, y el usuario podrá volver a “Eventos en...” para buscar otra ciudad:

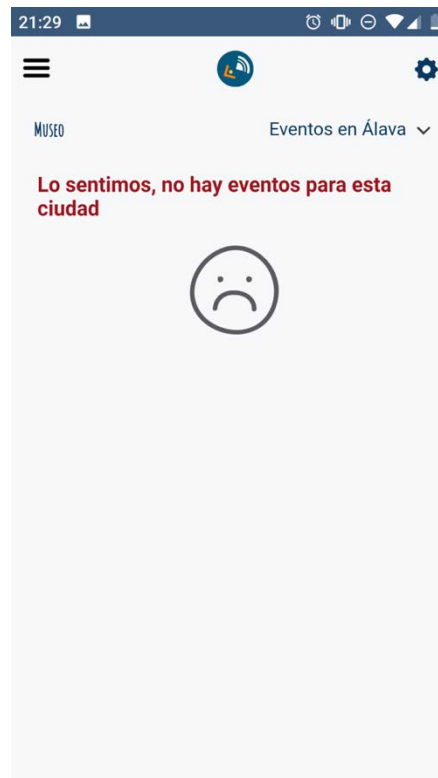


Ilustración 34: Ciudad sin eventos disponibles IU-13

5.4.3.1 Interfaces de modo invitado

A continuación, se va a mostrar la pantalla principal y los eventos del usuario invitado, es decir, cuando el usuario ha entrado sin registrarse:

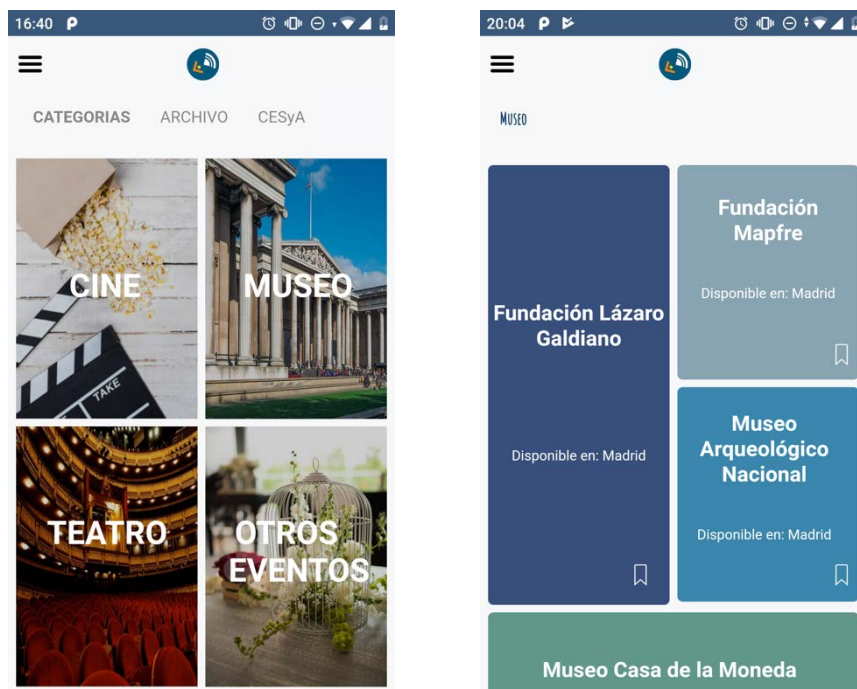


Ilustración 35: Inicio modo invitado IU – 14

Ilustración 36: Eventos en modo invitado IU – 16

Un usuario invitado puede acceder a Mi Archivo, para ver que ahí estarían los eventos favoritos si se registrase:

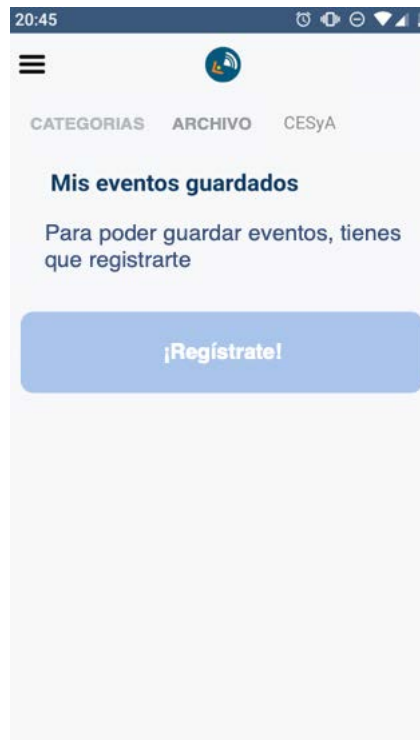


Ilustración 37: Mi Archivo Usuario invitado IU - 15

Si un usuario invitado quiere guardar un evento, dado a que no puede hacerlo, se le notificará de lo que debe hacer para guardarlo:

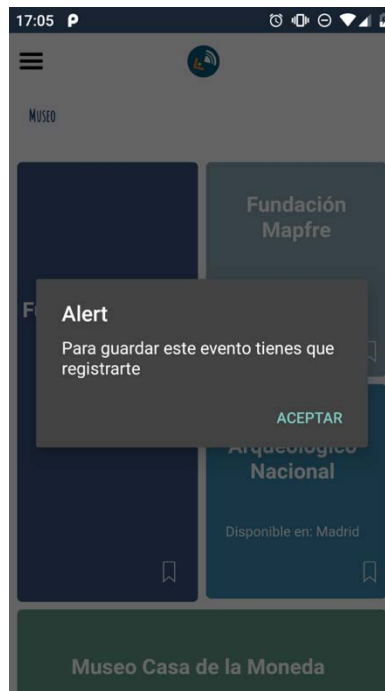


Ilustración 38: Guardar eventos usuario invitado IU -17

El usuario deberá registrarse si desea guardar el evento.

5.6.4 Interfaces de almacenamiento de datos

Ilustración 39: Configuración II CU- 18

Ilustración 40: Configuración IU - 18

La ventana de *Configuración* es una única pantalla, en la que se muestra al usuario la posibilidad de modificar los datos de su cuenta. Para minimizar la carga de memoria del usuario, se pone sobre los campos a rellenar, sus datos actuales. La contraseña no se muestra por seguridad. El botón “*Guardar cambios*” gestiona el cambio de esta información sobre la base de datos y vuelve a la pantalla principal.

Volviendo a los eventos de cada categoría, las interfaces de almacenamiento de datos tratan el indicador de evento favorito. El usuario puede pulsar sobre él, y de esta manera se almacena directamente en la base de datos, mostrándose en “*Mi Archivo*”. El icono queda relleno (así el usuario sabe que ya está marcado como favorito):



Ilustración 41: Evento favorito IU-19

Ilustración 42: Evento guardado en Mi Archivo IU-05

El usuario puede desmarcar de nuevo el evento como favorito, volviendo a quedarse el icono sin rellenar. El evento desmarcado desaparecería de Mi Archivo.

5.7 Consistencia de las interfaces de usuario

Para verificar el correcto funcionamiento y la calidad de las interfaces, se va a analizar la consistencia entre las interfaces y los casos de uso del sistema. Cada caso de uso debe estar cubierto por las interfaces de la aplicación.

Esta consistencia se va a mostrar en forma de tabla, que contendrá cada caso de uso con las interfaces de usuario que lo relacionen. Si existe consistencia entre ellos, se pondrá "X"; en caso contrario se dejará el espacio en blanco.

5.7.1 Acceso a la aplicación: Casos de uso - interfaces de usuario

	IU-00	IU-01	IU-02	IU-03	IU-14
CU-00	X				
CU-01					X
CU-02			X	X	
CU-03		X			

Tabla 108: Consistencia CU-IU Acceso a la aplicación

5.7.2 Navegación por la aplicación: Casos de uso – interfaces de usuario

	IU-04	IU-05	IU-06	IU-07	IU-08	IU-09	IU-10	IU-14	IU-16	IU-17
CU-04	X							X		
CU-05	X									
CU-06		X								
CU-07		X								
CU-08									X	
CU-09			X							
CU-10						X				
CU-11						X	X			X
CU-12							X			
CU-13						X				
CU-14							X			
CU-15										
CU-16				X						
CU-17				X	X					

CU-18				X	X					
CU-19				X	X					
CU-20			X							
CU-21			X	X						
CU-22	X									

Tabla 109: Consistencia CU-IU Navegación por la aplicación

5.7.3 Filtro de eventos: Casos de uso - interfaces de usuario

	IU-11	IU-12	IU-13	IU-14
CU-23	X	X		
CU-24		X		
CU-25			X	X
CU-26	X	X	X	
CU-27	X	X		
CU-28	X			X

Tabla 110: Consistencia CU-IU Filtro de eventos

5.7.4 Almacenamiento de datos: Casos de uso - interfaces de usuario

	IU-18	IU-19	IU-20	IU-05	IU-09
CU-29		X			
CU-30	X		X	X	
CU-31			X		X

Tabla 111: Consistencia CU-IU Almacenamiento de datos

6. DISEÑO DEL SISTEMA

6.1 Objetivo

Este apartado resolverá el problema que se ha descrito y modelado en el análisis del sistema. Se generarán las especificaciones de construcción relativas al mismo.

Para especificar este diseño, se detallará tanto la arquitectura del sistema como su comportamiento para comprender de forma total el funcionamiento y desarrollo de la aplicación.

6.2 Arquitectura del sistema

Este sistema presenta una arquitectura MVC, que es la estructura origen de la alternativa escogida para su desarrollo, Apache Córdoba.

El proyecto se va a desarrollar en un lenguaje orientado a prototipos, que conlleva un tratamiento diferente al lenguaje orientado a objetos. Esto implica que la arquitectura del sistema no esté estructurada en clases, y por tanto, no cuente con un modelo de clases ni con instancias de objetos.

Bien es cierto que desde las especificaciones de *EcmaScript 6* (19) se ha incluido la gestión de clases en JavaScript, pero en este proyecto se ha decidido seguir esta estructura ya que la orientación a prototipos es mucho más flexible. No obstante, se va a diseñar la relación que existe entre cada elemento de la arquitectura que sigue el sistema.

Cada prototipo será el encargado de gestionar el *controlador* del sistema, y contienen propiedades y variables para nutrir de lógica al resto de módulos de la arquitectura.

Tanto la *vista* como el *modelo* son gestionados por diferentes componentes del sistema.

Siguiendo, por tanto, la estructura MVC, cada una de las capas sigue esta definición:

6.2.1 Controlador

Esta capa lo gestionan los prototipos *JavaScript* y serán los encargados de proporcionar la lógica a la aplicación. Estos prototipos cuentan con propiedades y variables, que serán utilizadas por el resto de los componentes.

Además de éstos, hay ciertos componentes *HTML* que necesitan las propiedades de los prototipos al momento de su ejecución, por lo que formarían parte también del controlador.

A continuación, se muestra la capa *Controlador* con los prototipos y componentes que lo estructuran:

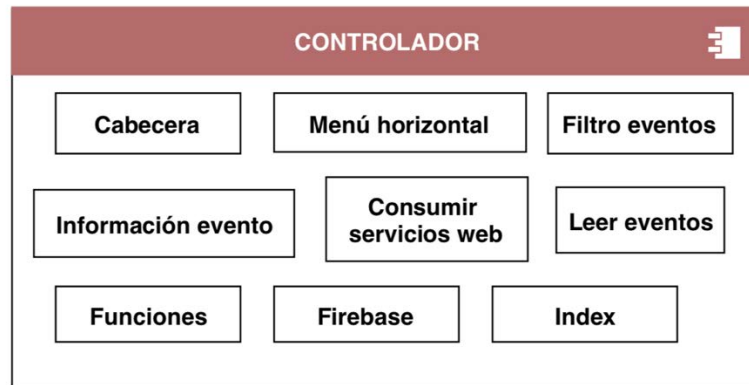


Ilustración 43: Capa controlador MVC

6.2.2 Vista

Esta capa lo gestionan los componentes HTML y CSS.

Los componentes HTML llevarán embebido el acceso a prototipos externos, creándose así la relación entre la *vista* y el *controlador*. Además, llevarán enlazados componentes externos CSS para proporcionar el estilo del sistema.

Los componentes HTML, por tanto, serán el resultado de la vista, lo que ve el usuario final. A continuación, se muestra la capa *Vista* con los componentes que lo forman:

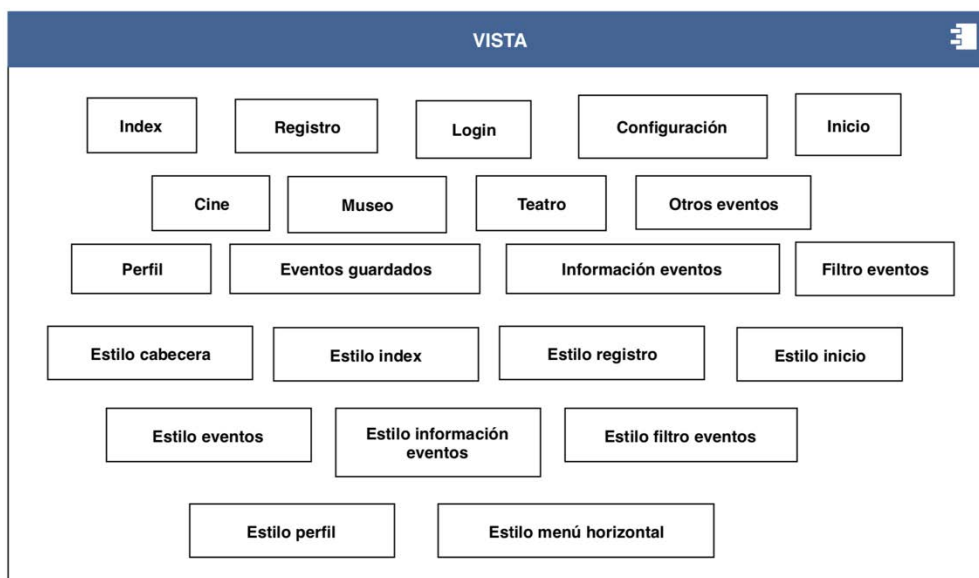


Ilustración 44: Capa vista MVC

6.2.3 Modelo

Esta capa es la encargada de nutrir al sistema de toda la información. Habrá dos conexiones diferentes para proveer los datos, como ya se ha visto anteriormente:

- Base de datos
- *Web Services* de CESyA

Los componentes que forman esta capa serán, por tanto:

- Tabla física “*Users*” de la base de datos del sistema
- Modelo de datos de los *Web Services* de CESyA (*es importante recordar que no se tiene el conocimiento de este modelo*)



Ilustración 45: Capa Modelo MVC

Teniendo ya la visión de cada capa, se va a realizar una síntesis para tener el contexto global de la arquitectura del sistema:

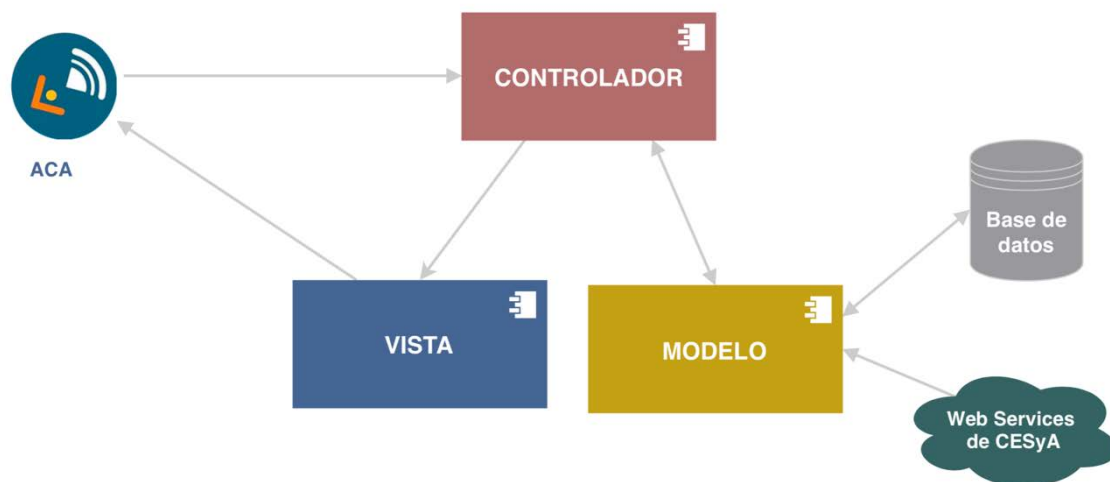


Ilustración 46: Arquitectura del sistema

El *controlador* gestiona todo el funcionamiento lógico de la aplicación. Éste conecta, de forma indirecta, mediante el modelo con los sistemas que tratan la información (la base de datos y los servicios web).

Una vez tiene esta información para construir la aplicación, se encarga de comunicárselo a la *vista*. La *vista* será quien muestre al usuario toda la información. No tiene conexión directa con

el *modelo* porque no lo necesita. Esta capa es la que proporciona las respuestas a las peticiones del usuario. Tampoco dirige ningún tipo de información al *controlador*, es una relación unidireccional. Su función simplemente es mostrar el sistema al usuario.

El *modelo* está en constante comunicación con el *controlador*. Como se ha explicado en apartados anteriores, se comunica tanto con la base de datos como con los servicios web de CESyA. La conexión a la base de datos es bidireccional, puede tanto leer como escribir en ella. Sin embargo, con los servicios web únicamente extrae su información, manteniendo una relación de una única dirección.

El usuario es quien realiza las peticiones al *controlador* para el funcionamiento de la aplicación. Por eso, el controlador recibe estas peticiones y las gestiona, comenzando el flujo que se acaba de explicar.

6.3 Interacción de prototipos y componentes

Este apartado va a describir la relación que existe entre la arquitectura del sistema con los casos de uso y las interfaces de usuario definidas en el *Análisis del Sistema*.

Los prototipos y los componentes, que son los elementos que forman la arquitectura del sistema, se definirán con más detalle en el apartado 4.4 y 4.5, respectivamente.

Los casos de uso interactúan con los prototipos del sistema porque son los encargados de cubrir la lógica que llevaría a cabo la realización de estos. Las interfaces de usuario interactúan con los componentes porque son los que gestionan la *vista* del sistema.

Estas interacciones se van a mostrar en forma de matriz de trazabilidad, respectivamente para cada elemento.

6.3.1 Interacción prototipos y casos de uso

A continuación, se muestra la trazabilidad entre los casos de uso y los prototipos del sistema:

	PT-00	PT-01	PT-02	PT-03	PT-04	PT-05	PT-06	PT-07	PT-08
CU-00	X					X			
CU-01						X			
CU-02		X				X			
CU-03		X				X			
CU-04	X					X			X
CU-05	X					X			X

CU-06	X					X			X
CU-07	X	X				X	X		X
CU-08	X					X			X
CU-09	X					X			X
CU-10	X	X				X			
CU-11	X	X				X			
CU-12	X	X				X	X		
CU-13				X	X				
CU-14				X	X		X		
CU-15	X	X	X				X	X	X
CU-16	X	X	X					X	X
CU-17						X			
CU-18				X	X				
CU-19						X	X		
CU-20									
CU-21						X			
CU-22	X	X	X					X	X

6.3.2 Interacción componentes e interfaces de usuario

A continuación, se muestra la trazabilidad entre las interfaces de usuario y los componentes:

	CT-00	CT-01	CT-02	CT-03	CT-04	CT-05	CT-06	CT-07	CT-08
IU-00	X					X			
IU-01						X			
IU-02		X				X			

IU-03		X				X			
IU-04						X			X
IU-05						X			X
IU-06						X			X
IU-07		X				X	X		X
IU-08						X			X
IU-09						X			X
IU-10		X				X			
IU-11		X				X			
IU-12		X				X	X		
IU-13				X	X				
IU-14				X	X		X		
IU-15		X	X				X	X	X
IU-16		X	X					X	X
IU-17						X			
IU-18				X	X				
IU-19						X	X		
IU-20									
IU-21						X			
IU-22	X	X	X					X	X

Tabla 112: Trazabilidad IU- CT 1

	CT-09	CT-10	CT-11	CT-12	CT-13	CT-14	CT-15	CT-16	CT-17
IU-00	X					X			
IU-01						X			
IU-02		X				X			

IU-03		X				X			
IU-04	X					X			X
IU-05	X					X			X
IU-06	X					X			X
IU-07	X	X				X	X		X
IU-08	X					X			X
IU-09	X					X			X
IU-10	X	X				X			
IU-11	X	X				X			
IU-12	X	X				X	X		
IU-13				X	X				
IU-14				X	X		X		
IU-15	X	X	X				X	X	X
IU-16	X	X	X					X	X
IU-17						X			
IU-18				X	X				
IU-19						X	X		
IU-20									
IU-21						X			
IU-22	X	X	X					X	X

Tabla 113: Trazabilidad IU- CT 2

	CT-18	CT-19	CT-20	CT-21	CT-22
IU-00	X				
IU-01					
IU-02		X			
IU-03		X			

IU-04	X				
IU-05	X				
IU-06	X				
IU-07	X	X			
IU-08	X				
IU-09	X				
IU-10	X	X			
IU-11	X	X			
IU-12	X	X			
IU-13				X	X
IU-14				X	X
IU-15	X	X	X		
IU-16	X	X	X		
IU-17					
IU-18				X	X
IU-19					
IU-20					
IU-21					
IU-22	X	X	X		

Tabla 114: Trazabilidad IU- CT 3

6.4 Diseño de prototipos

Siguiendo la arquitectura del sistema, se van a diseñar los prototipos que la definen. Estos objetos componen la capa *Controlador* del sistema, aportando el funcionamiento lógico de la aplicación.

6.4.1 Definición de los prototipos

Para explicar cada objeto prototipo, se expondrá una tabla como la siguiente:

ID	PT-XX	
NOMBRE		
VARIABLES		
FUNCIONES	<i>Nombre función</i>	<i>descripción</i>

Tabla 115: Definición de prototipos

Donde cada campo sigue esta descripción:

- **ID:** Identificador único del prototipo. Consta del siguiente formato:
 - PT: abreviatura de Prototipo
 - XX: número del 00 al 99
- **Nombre:** nombre del objeto prototipo
- **Variables:** posibles variables que tenga el prototipo. Si no tiene, estará vacío
- **Funciones:** métodos/funciones que tenga el prototipo
 - *Nombre función:* nombre de la función
 - *Descripción:* breve descripción del objetivo de la función

Cabecera

Este módulo se encarga de mostrar el menú lateral de las interfaces. Estará habilitado en casi todas las interfaces, como ya se definió en el análisis del sistema.

ID	PT-00	
NOMBRE	Cabecera	
VARIABLES		
FUNCIONES	<i>cerrar_tactil()</i>	Se encarga de cerrar el menú lateral tocando la parte sombreada de la pantalla o deslizando el menú hacia la izquierda
	<i>abrir_tactil()</i>	Se encarga de abrir el menú lateral moviendo el dedo sobre el borde izquierdo de la pantalla
	<i>abrir_boton()</i>	Se encarga de abrir el menú lateral tocando el icono de “menú lateral”
	<i>Cerrar_boton()</i>	Se encarga de cerrar el menú lateral desde el icono “X”

Ilustración 47: Prototipo Cabecera PT-00

Firestore

Este módulo se encarga de la gestión de información que proviene o que se va a almacenar en la base de datos del sistema. Las operaciones sobre la base de datos son:

- Insertar usuarios cuando se registren, su información y sus eventos favoritos
- Borrar eventos desmarcados de favoritos
- Actualizar información de un usuario
- Consultar toda la información

Para la conexión con la base de datos, en primer lugar, es necesario que todos los componentes HTML que requieran de éste lleven embebido las credenciales de autenticación a la base de datos.

En segundo lugar, hay que aclarar que se trata de una base de datos de *Google*, por lo que tiene una interfaz propia para la gestión de esta. Por tanto, para cada tipo de acción se han seguido estos pasos:

Insertar usuarios

Para hacer inserciones, esta base de datos utiliza diferentes métodos. El escogido para este sistema será el método *push()*, que agrega un elemento con identificador único a la lista de usuarios.

Insertar eventos de los usuarios

Para guardar los eventos que un usuario ha marcado como favorito, hay que acceder primero al usuario, y seguidamente al atributo correspondiente. Se guardará en un array los identificadores de cada uno de ellos.

Borrar información

Para borrar los eventos que un usuario marcó como favorito, hay que acceder primero al usuario, y seguidamente al evento correspondiente. Mediante el método *remove()* de Firestore se realiza esta acción.

Consultar información

Para realizar cualquier tipo de consulta sobre los datos, este sistema ha escogido el método *on()* de Firestore, que pasándole el identificador de la información que se quiera consultar, devuelve un objeto con dicha información.

Actualizar información

La acción de actualizar datos sobre la base de datos se realiza en primer lugar con el método `on()` de Firebase para consultar qué se quiere actualizar. Una vez se tiene esa información, se emplea el método `set()` que actualiza la información con los datos nuevos.

Este prototipo también se encargará de validar el acceso a la aplicación y de comunicar a la *vista* la información que provenga de la base de datos.

ID	PT-01	
NOMBRE	Firebase	
VARIABLES	<ul style="list-style-type: none"> • Num_usuarios • Id_user • Users [] • Contraseñas [] 	
FUNCIONES	<i>SHA1(msg)</i>	Encripta las contraseñas de los usuarios
	<i>go_firebase()</i>	Redirecciona el sistema a otra pantalla de la aplicación
	<i>registrar_usuario()</i>	Inserta la información del usuario en la base de datos cuando éste se registra
	<i>update_perfil()</i>	Modifica los datos de la cuenta de usuario cuando éste desea cambiarlos
	<i>guardar_evento(id_user, id_div, id_evento)</i>	Almacena los eventos que el usuario ha marcado como favorito
	<i>Leer_datos_usuario_cabecera()</i>	Provee a la cabecera de la aplicación los datos del usuario logueado
	<i>Leer_datos_usuario_editar()</i>	Provee a la pantalla Configuración de los datos del usuario logueado
	<i>Obtener_usuarios()</i>	Lista los usuarios que hay en la base de datos
	<i>Validar_login()</i>	Gestiona el acceso a la aplicación
	<i>Obtener_imagen_evento()</i>	Obtiene las imágenes de los eventos

<i>Mi_perfil()</i>	Provee a la pantalla Mi Perfil de los datos personales del usuario
<i>Mis_eventos()</i>	Muestra en Mi Archivo los eventos que el usuario tiene guardados

Ilustración 48: Prototipo Firebase PT-01

Consumir servicios Web

Este módulo es el encargado de realizar las llamadas al *Web Services* de CESyA para obtener la información en relación con los eventos de la aplicación.

Estas llamadas a los servicios web se realizarán mediante peticiones *XMLHttpRequest* con parámetro *POST* pasándole estas variables:

- **PROVINCIA** [Álava, Albacete, Alicante, Almería, Asturias, Ávila, Badajoz, Barcelona, Burgos, Cáceres, Cádiz, Santander, Castellón, Ceuta, Ciudad Real, Córdoba, A Coruña, Cuenca, Girona, Granada, Guadalajara, Guipúzcoa, Huelva, Huesca, Illes Balears, Jaén, León, Lleida, Lugo, Madrid, Málaga, Melilla, Murcia, Navarra, Ourense, Palencia, Las Palmas, Pontevedra, La Rioja, Salamanca, Santa Cruz de Tenerife, Segovia, Sevilla, Soria, Tarragona, Teruel, Toledo, Valencia, Valladolid, Vizcaya, Zamora, Zaragoza]
- **GENERO** [cine, teatro, museo, otros]
- **ACCESIBILIDAD** [subtitulado, accesibilidad completa, lenguaje de signos o audiodescripción]

Las respuestas a estas peticiones se retornan en formato JSON, devolviendo:

1. Cantidad de eventos en una provincia con un servicio de accesibilidad
2. Lista de obras para una provincia con un género y servicio de accesibilidad específico
3. Información de una obra concreta en una provincia con un género y servicio de accesibilidad específico

La información de cada respuesta depende de las variables que se pasen en la petición. Esto significa que no es información estática, cambia en base a los filtros aplicados. Para lograr un diseño dinámico en la *vista* se han seguido estas especificaciones:

1. Se define un patrón de 4 contenedores de eventos
2. Se divide el número total de eventos entre 4
 - a. Cociente: *c*
 - b. Resto: *r*
3. Si el cociente es mayor que 0, significa que hay más de 4 eventos:
 - a. Se dibuja *c* veces el patrón de los 4 eventos y se añaden *r* eventos
4. Si el cociente es menor que 0, significa que hay menos de 4 eventos:
 - a. Se dibujan la cantidad de eventos que haya para esa petición

Además, se ha tenido en cuenta el caso de que la petición no devuelva datos. Eso es debido a que para los filtros aplicados no habría resultados. Se muestra un mensaje de error alertando al usuario de la situación.

ID	PT-02	
NOMBRE	Consumir servicios Web	
VARIABLES		
FUNCIONES	<i>div_dinamico(obras,ids,loop,top)</i>	Dibuja el patrón de los 4 eventos de forma dinámica
	<i>no_eventos()</i>	Gestiona de forma dinámica la interfaz que avisa al usuario de que no hay eventos con los filtros aplicados
	<i>consumir_datos()</i>	Dibuja todos los eventos disponibles con los filtros aplicados

Tabla 116: Prototipo Consumir servicios Web PT-02

Filtro eventos

Este módulo gestiona el filtro que el usuario puede aplicar sobre los eventos.

De forma dinámica, se puede buscar cualquier provincia. Esta búsqueda consiste en comparar la cadena que el usuario ha introducido con cada una de las provincias, tanto con su nombre correcto como con todas sus posibles acepciones.

- Si la búsqueda da resultados, la ciudad está en el rango de provincias. Se muestra un contenedor con la ciudad para poder seleccionarla.
- Si la búsqueda no ha dado resultados, esa ciudad no está en el rango de provincias.

Para seleccionar el servicio de accesibilidad basta con escoger una de las cuatro opciones disponibles.

Este módulo, para comunicar al resto de ellos qué filtros se han seleccionado, guarda mediante *LocalStorage* estos valores. Así también se tiene un control sobre si el usuario ha seleccionado ambos filtros, ya que de no ser así no podría continuar.

ID	PT-03	
NOMBRE	Filtro eventos	
VARIABLES	<ul style="list-style-type: none"> • <i>Ciudad_usuario</i> • <i>Control_ciudad</i> • <i>Control_access</i> • <i>Ciudad_busqueda</i> • <i>Posición_img_ciudad</i> • <i>Selecc_ciudad</i> • <i>Param_ciudades[]</i> 	
FUNCIONES	<i>Buscar_ciudad()</i>	Busca la ciudad que el usuario ha introducido por parámetro
	<i>Seleccionar_accesibilidad()</i>	Selecciona el servicio de accesibilidad que el usuario haya escogido
	<i>seleccionar_ciudad()</i>	Selecciona la ciudad que ha escogido el usuario cuando es una de las populares
	<i>Aplicar_filtros()</i>	Aplica los filtros seleccionados por el usuario

Tabla 117: Prototipo Filtro eventos PT-03

Funciones

Este módulo gestiona diferentes funciones que son necesarias para la vista de la aplicación.

Se encarga de subir a *LocalStorage* el id del evento que el usuario seleccione para abrir.

ID	PT-04	
NOMBRE	Funciones	
VARIABLES		
FUNCIONES	<i>go (pagina)</i>	Redirecciona el sistema a una pantalla de la aplicación determinada
	<i>visualiza_segundoFormu()</i>	Visualiza la segunda parte del formulario del registro de usuario

	<i>cambiar_div()</i>	Muestr/ocupa contenedores
	<i>Cambiar(clase)</i>	Cambia de pestaña en la ventana principal de la aplicación
	<i>Mover_divInfo()</i>	Gestiona los contenedores del cambio pestaña en la ventana principal de la aplicación
	<i>Mover()</i>	Gestiona los contenedores del cambio pestaña en la ventana principal de la aplicación
	<i>NoGuardar_evento(id)</i>	Desmarca un evento favorito, dejándolo en su estado original
	<i>Cerrar_sesion()</i>	Cierra la sesión del usuario en ese dispositivo
	<i>Info_evento()</i>	Redirecciona a la pantalla de información de un evento

Tabla 118: Prototipo Funciones PT-04

Información evento

Este módulo proporciona a la *vista* la información de un evento concreto.

Para mostrar esta información, lee de *LocalStorage* los valores del evento que el usuario ha seleccionado (este almacenamiento en *LocalStorage* lo gestionan los módulos *Filtro_ciudad* y *Funciones*).

En base a esta información, realiza la petición al *Web Service* mediante XMLHttpRequest, con POST y pasándole estos parámetros:

- **Provincia:** ciudad que ha seleccionado el usuario
- **Género:** genero del evento seleccionado
- **Accesibilidad:** servicio de accesibilidad del evento seleccionado

ID	PT-05
NOMBRE	Información evento

VARIABLES		
FUNCIONES	<i>go (pagina)</i>	Redirecciona el sistema a una pantalla de la aplicación determinada
	<i>visualiza_segundoFormu()</i>	Visualiza la segunda parte del formulario del registro de usuario
	<i>cambiar_div()</i>	Muestr/ocupa contenedores
	<i>Cambiar(clase)</i>	Cambia de pestaña en la ventana principal de la aplicación
	<i>Mover_divInfo()</i>	Gestiona los contenedores del cambio pestaña en la ventana principal de la aplicación
	<i>Mover()</i>	Gestiona los contenedores del cambio pestaña en la ventana principal de la aplicación
	<i>NoGuardar_evento(id)</i>	Desmarca un evento favorito, dejándolo en su estado original
	<i>Cerrar_sesion()</i>	Cierra la sesión del usuario en ese dispositivo
	<i>Info_evento()</i>	Redirecciona a la pantalla de información de un evento

Tabla 119: Prototipo Información evento PT-05

Leer eventos

Este prototipo tiene la función de leer los eventos que el usuario ha guardado. Los mostrará en Mi Archivo.

ID	PT-06	
NOMBRE	Leer evento	
VARIABLES	<ul style="list-style-type: none"> • <i>Cantidad_eventos</i> • <i>Cantidad_eventos_json</i> • <i>Info_obra</i> • <i>Info_obra_json</i> 	
FUNCIONES	<i>div_dinamico(obras,ids,loop,top)</i>	Dibuja el patrón de los 4 eventos de forma dinámica

	<i>Mostrar_eventos()</i>	Gestiona de forma dinámica los eventos que el usuario ha guardado
	<i>Info_evento()</i>	Muestra la información del evento

Tabla 120: Prototipo Leer eventos PT-06

Menú horizontal

Este módulo gestiona la lógica del menú horizontal que se encuentra en la ventana principal de la aplicación.

ID	PT-07	
NOMBRE	Menú horizontal	
VARIABLES	<ul style="list-style-type: none"> • <i>Settings</i> • <i>colores</i> 	
FUNCIONES	<i>Determinar_disposicion (contenido, contenedor)</i>	Gestiona las posiciones de los contenedores de la información
	<i>Mover_pestaña()</i>	Gestiona el movimiento de las pestañas cuando se pulsa en una u en otra

Tabla 121: Prototipo Menú horizontal PT-07

6.5 Diseño de componentes

Una vez explicados los prototipos que componen el *Controlador* de la arquitectura del sistema, se van a diseñar el resto de los componentes que gestionan la aplicación. Estos componentes forman parte de la *Vista* y del *Modelo* de la arquitectura.

6.5.1 Definición de los componentes

Para la especificación de estos componentes se va a seguir esta tabla:

ID	CT-XX	
NOMBRE		
DESCRIPCIÓN		
FUNCIONES	<i>Nombre función</i>	<i>Descripción</i>

VARIABLES	
CAPA	

Donde cada campo sigue esta definición:

- **ID:** Identificador único del prototipo. Consta del siguiente formato:
 - CT: abreviatura de Componente
 - XX: número del 00 al 99
- **Nombre:** nombre del componente.
- **Descripción:** breve descripción del objetivo del componente.
- **Funciones:** posibles funciones que pueda necesitar el componente. Al no ser la capa Controlador, no todos los componentes tendrán funciones, por lo que este campo estará vacío en ese caso.
 - *Nombre función:* nombre de la función
 - *Descripción:* breve descripción del objetivo de la función
- **Variables:** posibles variables que pueda tener el componente. Si no tiene, este campo estará vacío.
- **Capa:** capa de la arquitectura a la que pertenece. Puede ser o *Vista* o *Modelo*

A continuación, se detalla la definición de cada componente:

ID	CT-00	
NOMBRE	Index	
DESCRIPCIÓN	Muestra la pantalla inicial de la aplicación	
FUNCIONES	<i>Iniciar()</i>	Gestiona el inicio de la aplicación, controlando los datos de sesión
VARIABLES	<ul style="list-style-type: none"> • <i>Config:</i> almacena las credenciales de autenticación de la base de datos 	
CAPA	Vista	

Tabla 122: Componente Index CT-00

ID	CT-01	
NOMBRE	Registro	
DESCRIPCIÓN	Muestra el registro del usuario en el sistema	
FUNCIONES		
VARIABLES	<ul style="list-style-type: none"> • <i>Config:</i> almacena las credenciales de autenticación de la base de datos 	

CAPA	<i>Vista</i>
-------------	--------------

Tabla 123: Componente Registro CT-01

ID	CT-02
NOMBRE	Login
DESCRIPCIÓN	Muestra el inicio de sesión del usuario a la aplicación
FUNCIONES	
VARIABLES	<ul style="list-style-type: none"> • <i>Config</i>: almacena las credenciales de autenticación de la base de datos
CAPA	<i>Vista</i>

Tabla 124: Componente Login CT-02

ID	CT-03
NOMBRE	Inicio
DESCRIPCIÓN	Muestra la pantalla principal de la aplicación
FUNCIONES	
VARIABLES	<ul style="list-style-type: none"> • <i>Config</i>: almacena las credenciales de autenticación de la base de datos
CAPA	<i>Vista</i>

Tabla 125: Componente Inicio CT-03

ID	CT-04
NOMBRE	Inicio sin registrar
DESCRIPCIÓN	Muestra la pantalla principal de la aplicación para los usuarios invitados
FUNCIONES	
VARIABLES	<ul style="list-style-type: none"> • <i>Config</i>: almacena las credenciales de autenticación de la base de datos
CAPA	<i>Vista</i>

Tabla 126: Componente Inicio Sin Registrar CT-04

ID	CT-05
-----------	--------------

NOMBRE	Eventos sin registrar
DESCRIPCIÓN	Muestra los eventos para los usuarios invitados
FUNCIONES	
VARIABLES	<ul style="list-style-type: none"> • <i>Config</i>: almacena las credenciales de autenticación de la base de datos
CAPA	Vista

Tabla 127: Componente Eventos Sin Registrar CT-05

ID	CT-06
NOMBRE	Configuración
DESCRIPCIÓN	Muestra la gestión de cambio de datos de la cuenta de usuario
FUNCIONES	
VARIABLES	<ul style="list-style-type: none"> • <i>Config</i>: almacena las credenciales de autenticación de la base de datos
CAPA	Vista

Tabla 128: Componente Configuración CT-06

ID	CT-07
NOMBRE	Perfil
DESCRIPCIÓN	Muestra el perfil de la cuenta de usuario
FUNCIONES	
VARIABLES	<ul style="list-style-type: none"> • <i>Config</i>: almacena las credenciales de autenticación de la base de datos
CAPA	Vista

Tabla 129: Componente Perfil CT-07

ID	CT-08
NOMBRE	Eventos guardados
DESCRIPCIÓN	Muestra los eventos que el usuario marcó como favorito
FUNCIONES	

VARIABLES	<ul style="list-style-type: none"> <i>Config</i>: almacena las credenciales de autenticación de la base de datos
CAPA	<i>Vista</i>

Tabla 130: Componente Eventos guardados CT-08

ID	CT-09
NOMBRE	Eventos
DESCRIPCIÓN	Muestra los eventos que ofrece la aplicación
FUNCIONES	
VARIABLES	<ul style="list-style-type: none"> <i>Config</i>: almacena las credenciales de autenticación de la base de datos
CAPA	<i>Vista</i>

Tabla 131: Componente Eventos CT-09

ID	CT-10
NOMBRE	Información eventos
DESCRIPCIÓN	Muestra la información de un evento concreto
FUNCIONES	
VARIABLES	<ul style="list-style-type: none"> <i>Config</i>: almacena las credenciales de autenticación de la base de datos
CAPA	<i>Vista</i>

Tabla 132: Componente Información Eventos CT-10

ID	CT-11
NOMBRE	Filtro eventos
DESCRIPCIÓN	Muestra los filtros que el usuario puede aplicar sobre los eventos
FUNCIONES	
VARIABLES	<ul style="list-style-type: none"> <i>Config</i>: almacena las credenciales de autenticación de la base de datos
CAPA	<i>Vista</i>

Tabla 133: Componente Filtro Eventos CT-11

ID	CT-12
NOMBRE	Estilo cabecera
DESCRIPCIÓN	Gestiona el estilo de la cabecera de la aplicación
FUNCIONES	
VARIABLES	
CAPA	Vista

Tabla 134: Componente Estilo Cabecera CT-12

ID	CT-13
NOMBRE	Estilo eventos
DESCRIPCIÓN	Gestiona el estilo de los eventos que muestra la aplicación
FUNCIONES	
VARIABLES	
CAPA	Vista

Tabla 135: Componente Estilo Eventos CT-13

ID	CT-14
NOMBRE	Estilo filtro eventos
DESCRIPCIÓN	Gestiona el estilo de los filtros que puede aplicar el usuario sobre los eventos
FUNCIONES	
VARIABLES	
CAPA	Vista

Tabla 136: Componente Estilo Filtro Eventos CT-14

ID	CT-15
NOMBRE	Estilo index

DESCRIPCIÓN	Gestiona el estilo de la pantalla inicial de la aplicación
FUNCIONES	
VARIABLES	
CAPA	<i>Vista</i>

Tabla 137: Componente Estilo Index CT-15

ID	CT-16
NOMBRE	Estilo información eventos
DESCRIPCIÓN	Gestiona el estilo de la información de cada evento
FUNCIONES	
VARIABLES	
CAPA	<i>Vista</i>

Tabla 138: Componente Estilo Información Eventos CT-16

ID	CT-17
NOMBRE	Estilo Inicio
DESCRIPCIÓN	Gestiona el estilo de la pantalla principal de la aplicación
FUNCIONES	
VARIABLES	
CAPA	<i>Vista</i>

Tabla 139: Componente Estilo Inicio CT-17

ID	CT-18
NOMBRE	Estilo menú horizontal
DESCRIPCIÓN	Gestiona el estilo del menú horizontal existente en ciertas pantallas de la aplicación
FUNCIONES	
VARIABLES	

CAPA	<i>Vista</i>
-------------	--------------

Tabla 140: Componente Estilo Menú Horizontal CT-18

ID	CT-19
NOMBRE	Estilo perfil
DESCRIPCIÓN	Gestiona el estilo del perfil del usuario
FUNCIONES	
VARIABLES	
CAPA	<i>Vista</i>

Tabla 141: Componente Estilo Perfil CT-19

ID	CT-20
NOMBRE	Estilo registro
DESCRIPCIÓN	Gestiona el estilo del registro del usuario en el sistema
FUNCIONES	
VARIABLES	
CAPA	<i>Vista</i>

Tabla 142: Componente Estilo Registro CT-20

ID	CT-21
NOMBRE	Usuarios
DESCRIPCIÓN	Gestiona los usuarios de la base de datos
FUNCIONES	
VARIABLES	
CAPA	<i>Modelo</i>

Tabla 143: Componente Usuarios CT-21

ID	CT-22
-----------	--------------

NOMBRE	Servicios Web
DESCRIPCIÓN	Gestiona la información de eventos que se obtiene de los servicios web de CESyA
FUNCIONES	
VARIABLES	
CAPA	<i>Modelo</i>

Tabla 144: Componente Servicios Web CT-22

6.6 Relación entre capas de arquitectura

En este apartado se va a detallar la relación que existe entre los componentes y prototipos de cada capa.

En los apartados anteriores se ha especificado la relación existente entre las capas de la arquitectura del sistema, pero ahora se va a definir una relación con más detalle, viendo las dependencias entre los prototipos y los componentes de los distintos módulos.

Esta especificación de las relaciones se va a diseñar a partir de los módulos que componen la arquitectura del sistema.

Para cada uno de los puntos siguientes, se ha dividido la trazabilidad en varias tablas para facilitar su comprensión.

6.6.1 Relación Controlador y Vista

A continuación, se muestra la relación entre las capas del *controlador* y la *vista*. Para el *controlador* se tratan los prototipos y para la *vista*, los componentes.

Hay ciertos componentes incluidos en las tablas que no se relacionan con los prototipos, ya que solo aportan estilo a la interfaz, alejado de la lógica de la aplicación. Estos son aquellos que no tengan ningún prototipo relacionado.

	CT-00	CT-01	CT-02	CT-03	CT-04	CT-05	CT-06	CT-07
PT-00				X	X	X	X	
PT-01		X	X	X			X	X
PT-02						X		
PT-03								

PT-04	X	X	X	X	X	X	X	X
PT-05								
PT-06								
PT-07				X	X			

Tabla 145: Relación Controlador – Vista I

	CT-08	CT-09	CT-10	CT-11	CT-12	CT-14	CT-15	CT-16
PT-00	X	X	X	X	X			
PT-01								
PT-02	X	X	X	X				
PT-03				X		X		
PT-04	X	X	X	X				
PT-05			X			X		
PT-06	X							X
PT-07								

Tabla 146: Relación Controlador - Vista II

	CT-17	CT-18	CT-19	CT-20
PT-00				
PT-01				
PT-02				
PT-03				
PT-04				
PT-05				
PT-06				
PT-07				

Tabla 147: Relación Controlador - Vista III

6.6.2 Relación Controlador y Modelo

Este apartado muestra la relación entre el *controlador* y el *modelo*. Solo se muestran aquellos prototipos que estén relacionados con los componentes del *modelo*.

Aquellos prototipos que se encuentren sin enlazar con los componentes, es porque no presentan relación con el modelo.

	PT-00	PT-01	PT-02	PT-03	PT-04	PT-05	PT-06	PT-07
CT-21		X						
CT-22			X	X		X	X	

Tabla 148: Relación Controlador – Modelo

6.6.3 Relación Vista y Modelo

Este apartado muestra la relación entre los componentes de las diferentes capas, proviniendo del *modelo* y de la *vista*. No todos los componentes de la vista tienen que estar relacionados con los componentes del modelo, por lo que no habría trazabilidad entre ellos.

	CT-00	CT-01	CT-02	CT-03	CT-04	CT-05	CT-06	CT-07
CT-21		X	X	X	X		X	X
CT-22						X		

Tabla 149: Relación Vista – Modelo I

	CT-08	CT-09	CT-10	CT-11	CT-12	CT-13	CT-14	CT-15
CT-21	X							
CT-22	X	X	X	X		X	X	

Tabla 150: Relación Vista – Modelo II

	CT-16	CT-17	CT-18	CT-19	CT-19
CT-21				X	
CT-22	X				

Tabla 151: Relación Vista - Modelo III

6.6.4 Relación Vista

La capa vista de la arquitectura presenta muchos componentes, que están relacionados entre ellos. Para mostrar estas relaciones, se ha diseñado un diagrama que muestra de forma más grafica esta relación, que en formato tabular podría dificultar entendimiento.

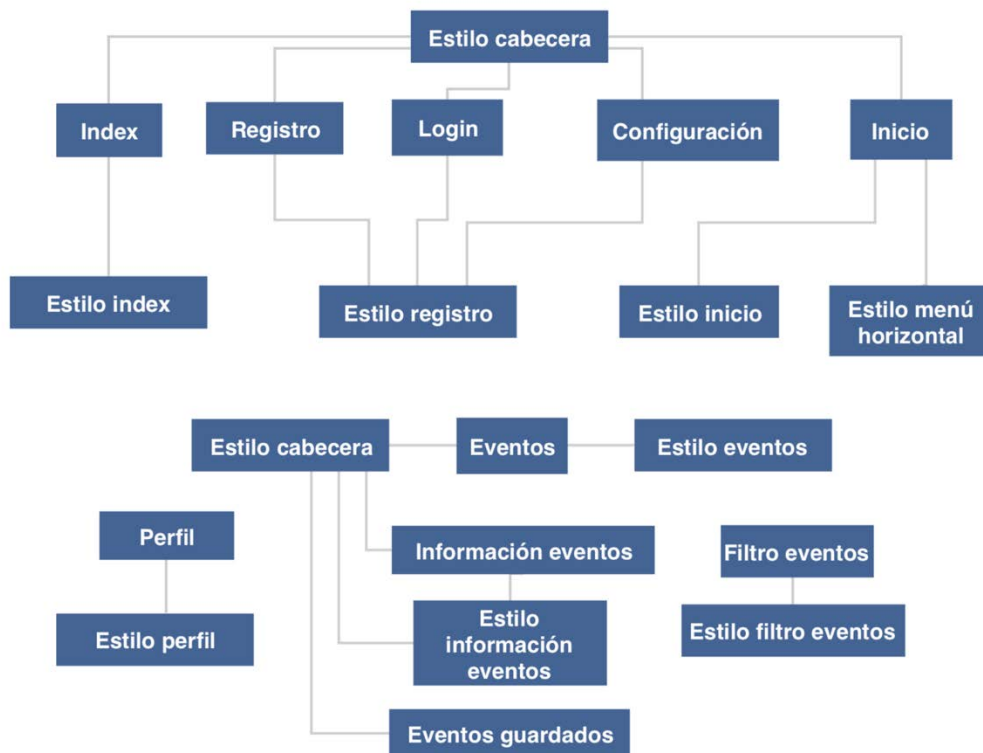


Ilustración 49: Relación entre componentes Vista

6.7 Diseño físico de datos

El diseño del modelo de la base de datos ya se estableció en el *Análisis del Sistema*. Este apartado diseñará la estructura física que va a seguir la base de datos.

Ya anteriormente se mencionó que la base de datos del sistema es no relacional. Contienen únicamente una colección, *Users*, donde se almacenan los usuarios que se registran en el sistema. Este proyecto no necesita guardar más información.

Esta colección tiene una relación directa con la información que proporcionan los *Web Services*, porque para cada usuario, se puede guardar los identificadores de los eventos que ha marcado como favoritos.

Para comprender de mejor manera esta relación conceptual, se ha diseñado un diagrama entidad-relación donde se muestra la dependencia y cardinalidad de estas entidades. La base de datos del sistema no es relacional por lo que técnicamente no sigue el paradigma de entidad-relación, propio de los modelos relacionales. Tampoco se dispone físicamente de la entidad *evento*, porque la proporciona los servicios web de CESyA, los cuales no sabemos qué modelo siguen. No obstante, este diagrama conceptual refleja la relación que existe y el funcionamiento de la gestión de los datos del sistema.

A continuación, se muestra dicho diagrama:

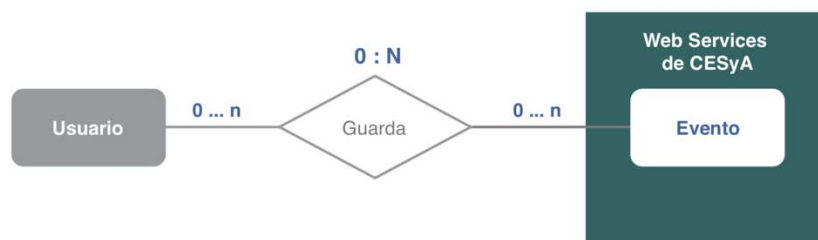


Ilustración 50: Diagrama E-R Usuarios y Eventos

Este diagrama expresa que un usuario puede tener desde ningún evento guardado, a tener los que quiera. De igual manera que un evento puede ser guardado desde ningún usuario hasta todos los que haya.

Además, el diagrama se traduciría como la creación de varias tablas físicas en la base de datos, pero siguiendo lo explicado anteriormente, solo existe en nuestro sistema la entidad Usuario, que es la equivalente colección *Users* de la base de datos.

El método que sigue la base de datos para almacenar es el método *clave-valor*. Esto quiere decir que, para cada campo de la entidad, guardar pares de información. La especificación que se va a seguir para guardar los eventos de cada usuario es la siguiente:

Ciudad - Servicio Accesible – ID del evento

De esta manera se tiene guardado qué evento se ha almacenado y también a qué ciudad y servicio accesible pertenece, para posteriormente poder trazarlo.

La entidad de la base de datos sigue esta estructura:

COLECCIÓN	<i>Users</i>
CAMPOS	<ul style="list-style-type: none"> • <i>email</i> : valor • <i>eventos_guardados</i> : <ul style="list-style-type: none"> ○ <i>Num_aleatorio</i> : Ciudad – Servicio Accesible – ID • <i>id</i> : valor • <i>login</i> : valor • <i>name</i> : valor • <i>password</i> : valor encriptado • <i>usuario</i> : valor

Tabla 152: Colección Users

Las contraseñas de los usuarios se almacenan de forma encriptada mediante el método SHA1.

Es importante aclarar que se trata de una base de datos en tiempo real, lo que implica que la gestión de la misma necesita conexión a internet. No obstante, en caso de que esta conexión al sistema de almacenamiento no estuviera disponible, tiene la capacidad de guardar los posibles cambios que la aplicación ha intentado hacer, y los retoma cuando vuelva la conexión.

Todo el sistema de este proyecto requiere de conexión a internet, por lo que este caso solo se daría si la base de datos está caída o se ha perdido el enlace a la misma.

7. VALIDACIÓN DEL SISTEMA

Este apartado valida el correcto funcionamiento del sistema. Para ello se va a definir un plan de pruebas, comprobando cada parte de la arquitectura del sistema y viendo que funciona como se espera.

7.1 Definición del plan de pruebas

El plan de pruebas se va a definir siguiendo esta tabla:

ID	PXX-NN
Objetivo	
Precondición	
Acción	
Resultado esperado	
Resultado obtenido	
Postcondición	

Tabla 153: Definición plan de pruebas

Cada campo de la tabla sigue esta descripción:

- **ID:** identificador de la prueba. Tiene este formato PXX-NN:
 - PXX: indica el tipo de prueba
 - NN: dígitos numéricos del 00 al 99
- **Precondición:** estado del sistema antes de que la prueba se realice
- **Acción:** tarea que realiza la prueba
- **Resultado esperado:** resultado que se espera obtener de la prueba, pudiendo ser *correcto* o *error*
- **Resultado obtenido:** resultado final de la prueba, pudiendo ser *correcto* o *error*
- **Postcondición:** estado del sistema tras realizarse la prueba, describiendo el resultado

Para realizar este plan de pruebas de manera ordenada, se van a separar las pruebas en base a la funcionalidad de la aplicación. Ya que en el *Análisis del Sistema* se definieron subsistemas basados en el funcionamiento del sistema, se usarán para esta clasificación. Por tanto, el campo identificador de la prueba puede tomar estos valores:

- **PAA:** Prueba de Acceso a la Aplicación
- **PNA:** Prueba de Navegación por la Aplicación
- **PFE:** Prueba de Filtro de Eventos

- PAD: Prueba de Almacenamiento de Datos

7.1.1 Pruebas de acceso a la aplicación

ID	PAA-00
Objetivo	Desplegar la aplicación en sistema operativo <i>Android</i>
Precondición	La aplicación no está desplegada en un dispositivo <i>Android</i>
Acción	Se implanta la aplicación en un dispositivo que tenga sistema operativo <i>Android</i>
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	La aplicación se despliega correctamente en <i>Android</i>

Tabla 154: PAA-00

ID	PAA-01
Objetivo	Desplegar la aplicación en sistema operativo <i>IOS</i>
Precondición	La aplicación no está desplegada en un dispositivo <i>IOS</i>
Acción	Se implanta la aplicación en un dispositivo que tenga sistema operativo <i>IOS</i>
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	La aplicación se despliega correctamente en <i>IOS</i>

Tabla 155: PAA-01

ID	PAA-02
Objetivo	Comprobar el correcto enlace entre capa <i>Controlador</i> y <i>Modelo</i>
Precondición	Un usuario que ha entrado a la aplicación aún no está registrado
Acción	Registrar el usuario en la base de datos del sistema

Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	El usuario se ha registrado correctamente en la aplicación, creándose una cuenta de usuario con los datos personales que ha introducido en el formulario

Tabla 156: PAA-02

ID PAA-03

Objetivo	Verificar que no se puede escoger un nombre de usuario que ya exista en el sistema
Precondición	Un usuario que ha entrado en la aplicación accede al formulario de registro
Acción	El usuario introduce un nombre de usuario, siendo conocedor de que ya existe en el sistema
Resultado esperado	Error
Resultado obtenido	Error
Postcondición	El sistema avisa al usuario de que ese nombre de usuario ya está registrado en el sistema

Tabla 157: PAA-03

ID PAA-04

Objetivo	Validar la contraseña del usuario antes de registrarlo en el sistema
Precondición	Un usuario que ha entrado en la aplicación accede al formulario de registro
Acción	El usuario repite la contraseña que ha elegido para proteger su cuenta
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	El usuario ha tenido que volver a introducir su contraseña para validar que la recuerda

Tabla 158: PAA-04

ID PAA-05

Objetivo	Comprobar que la contraseña del usuario es alfanumérica
Precondición	Un usuario que ha entrado en la aplicación accede al formulario de registro
Acción	Introducir una contraseña que solo tenga letras
Resultado esperado	Error
Resultado obtenido	Error
Postcondición	El sistema exige al usuario que introduzca una contraseña que contenga tanto letras como números

Tabla 159: PAA-05

ID PAA-06

Objetivo	Comprobar el carácter obligatorio de los campos del formulario de registro
Precondición	Un usuario que ha entrado en la aplicación accede al formulario de registro
Acción	Dejar vacío algún campo que el formulario indique que es obligatorio
Resultado esperado	Error
Resultado obtenido	Error
Postcondición	El sistema exige al usuario que complete todos los campos marcados con asterisco, es decir, los necesarios

Tabla 160: PAA-06

ID PAA-07

Objetivo	Comprobar el carácter opcional del campo imagen del formulario de registro
Precondición	Un usuario que ha entrado en la aplicación accede al formulario de registro
Acción	No adjuntar ninguna imagen en el campo imagen de perfil del formulario de registro

Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	El sistema permite correctamente al usuario registrarle en el sistema sin haber incluido imagen de perfil

Tabla 161: PAA-07

ID PAA-08

Objetivo	Comprobar el funcionamiento de la aplicación con conexión a internet
Precondición	Un usuario ha descargado la aplicación
Acción	Un usuario entra en la aplicación, teniendo su dispositivo conectado a internet
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	El sistema se despliega perfectamente

Tabla 162: PAA-08

ID PAA-09

Objetivo	Comprobar el funcionamiento de la aplicación sin conexión a internet
Precondición	Un usuario ha descargado la aplicación
Acción	Un usuario entra en la aplicación, teniendo su dispositivo sin conexión a internet
Resultado esperado	Error
Resultado obtenido	Error
Postcondición	El sistema no puede desplegarse si no existe conexión a internet

Tabla 163: PAA-09

ID PAA-10

Objetivo	Comprobar que un usuario registrado puede iniciar sesión en la aplicación
Precondición	Un usuario se ha registrado en la aplicación
Acción	Un usuario inicia sesión
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	El usuario ha podido iniciar sesión en la aplicación con su cuenta

Tabla 164: PAA-10

ID PAA-11

Objetivo	Comprobar que un usuario invitado no puede iniciar sesión en la aplicación
Precondición	Un usuario ha entrado en la aplicación
Acción	Un usuario no registrado intenta iniciar sesión
Resultado esperado	Error
Resultado obtenido	Error
Postcondición	El usuario no puede iniciar sesión porque no se ha registrado previamente

Tabla 165: PAA-11

ID PAA-12

Objetivo	Verificar que el usuario introduce su contraseña correcta al iniciar sesión
Precondición	Un usuario ha entrado en la aplicación
Acción	Un usuario registrado inicia sesión con una contraseña incorrecta
Resultado esperado	Error
Resultado obtenido	Error

Postcondición

El usuario no puede iniciar sesión porque la contraseña introducida no es válida

Tabla 166: PAA-12

ID**PAA-13**

Objetivo	Comprobar que un usuario invitado no puede acceder a la aplicación como usuario registrado
Precondición	Un usuario ha entrado en la aplicación
Acción	Un usuario no registrado intenta iniciar sesión
Resultado esperado	Error
Resultado obtenido	Error
Postcondición	El usuario no puede acceder a la aplicación como registrado porque no ha podido iniciar sesión

Tabla 167: PAA-13

ID**PAA-14**

Objetivo	Comprobar que un usuario registrado puede acceder a la aplicación como usuario invitado
Precondición	Un usuario ha entrado en la aplicación
Acción	Un usuario registrado entra en la aplicación desde “Entrar sin registrarse”, habiendo cerrado su sesión
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	Un usuario registrado puede acceder en modo invitado porque ha cerrado su sesión y el sistema no sabe que sí tiene una cuenta

Tabla 168: PAA-14

ID**PAA-15**

Objetivo	Comprobar que un usuario que no se ha registrado sí puede entrar en la aplicación
Precondición	Un usuario ha entrado en la aplicación

Acción	Un usuario accede a a aplicación desde “Entrar sin registrarse”
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	El usuario puede entrar a la aplicación sin haberse tenido que crear una cuenta, con servicios restringidos

Tabla 169: PAA-15

ID PAA-16

Objetivo	Comprobar que un usuario invitado puede registrarse una vez ha entrado en la aplicación en modo invitado
Precondición	Un usuario ha entrado en la aplicación en modo invitado
Acción	Un usuario invitado accede a “Registrarse” desde el menú lateral
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	Un usuario que previamente no se registró, ha podido registrarse más adelante desde el modo invitado

Tabla 170: PAA-16

ID PAA-17

Objetivo	Comprobar que al iniciar sesión se accede a la pantalla principal de la aplicación
Precondición	Un usuario se ha entrado en la aplicación
Acción	Un usuario se loguea en la aplicación
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	El usuario accede a la pantalla principal tras iniciar sesión

Tabla 171: PAA-17

ID PAA-18

Objetivo	Comprobar que la aplicación puede descargarse desde Play Store
Precondición	Un usuario no tiene la aplicación en su dispositivo Android
Acción	Un usuario accede a Play Store y busca la aplicación ACA
Resultado esperado	Correcto
Resultado obtenido	Error
Postcondición	El usuario no puede descargarse la aplicación desde Play Store porque el proyecto no se ha implementado de forma real

Tabla 172: PAA-18

ID PAA-19

Objetivo	Comprobar que la aplicación puede descargarse desde App Store
Precondición	Un usuario no tiene la aplicación en su dispositivo IOS
Acción	Un usuario accede a App Store y busca la aplicación ACA
Resultado esperado	Correcto
Resultado obtenido	Error
Postcondición	El usuario no puede descargarse la aplicación desde App Store porque el proyecto no se ha implementado de forma real

Tabla 173: PAA-19

7.1.2 Pruebas de navegación por la aplicación

ID PNA-00

Objetivo	Comprobar que el catálogo de eventos tiene una imagen por cada género
Precondición	Un usuario ha entrado en la aplicación
Acción	Un usuario accede a la pestaña “Categorías”
Resultado esperado	Correcto

Resultado obtenido	Correcto
Postcondición	El catálogo de categorías muestra una imagen por cada género

Tabla 174: PNA-00

ID PNA-01

Objetivo	Comprobar que las capas Vista y Modelo están bien enlazadas
Precondición	Un usuario se ha logueado en la aplicación
Acción	El usuario accede a los eventos que guardó como favoritos en la pestaña “Mi Archivo”
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	El usuario ha accedido a sus eventos favoritos. Existe buena conexión entre la <i>vista</i> y el <i>modelo</i>

Tabla 175: PNA-01

ID PNA-02

Objetivo	Comprobar que la aplicación aporta información de CESyA
Precondición	Un usuario ha entrado en la aplicación
Acción	El usuario accede a la pestaña CESyA
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	La aplicación muestra información sobre CESyA

Tabla 176: PNA-02

ID PNA-03

Objetivo	Comprobar que el menú lateral se puede cerrar deslizando hacia la izquierda en la pantalla
Precondición	Un usuario ha entrado en la aplicación

Acción	El usuario cierra el menú lateral deslizándolo hacia la izquierda
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	El menú lateral se ha cerrado

Tabla 177: PNA-03

ID PNA-04

Objetivo	Comprobar que el menú lateral se puede cerrar desde el icono "X"
Precondición	Un usuario ha entrado en la aplicación
Acción	El usuario cierra el menú lateral pulsando en el icono "X"
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	El menú lateral se ha cerrado

Tabla 178: PNA-04

ID PNA-05

Objetivo	Comprobar que el menú lateral tiene acceso al perfil del usuario
Precondición	Un usuario está en el menú lateral
Acción	El usuario accede a Mi Perfil desde el menú lateral
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	El usuario puede acceder a su perfil desde el menú lateral

Tabla 179: PNA-05

ID PNA-06

Objetivo	Comprobar que el menú lateral tiene acceso a sus eventos guardados
Precondición	Un usuario está en el menú lateral
Acción	El usuario accede a sus eventos guardados desde el menú lateral
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	El usuario puede acceder a sus eventos guardados desde el menú lateral

Tabla 180: PNA-06

ID PNA-07

Objetivo	Comprobar que el usuario puede cerrar sesión desde su perfil
Precondición	Un usuario está en su perfil
Acción	El usuario cierra sesión desde su perfil
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	El usuario ha podido salir de la aplicación desde su perfil

Tabla 181: PNA-07

ID PNA-08

Objetivo	Comprobar que el usuario puede cerrar sesión desde el menú lateral
Precondición	Un usuario está en el menú lateral
Acción	El usuario cierra sesión desde el menú lateral
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	El usuario ha podido salir de la aplicación desde el menú lateral

Tabla 182: PNA-08

ID PNA-09

Objetivo	Comprobar que la aplicación muestra los eventos de una ciudad y un servicio accesible
Precondición	Un usuario ha entrado en la aplicación
Acción	El usuario accede un género de las cuatro categorías y entra en esa pestaña
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	La aplicación muestra los eventos de Madrid con accesibilidad completa

Tabla 183: PNA-09

ID PNA-10

Objetivo	Comprobar que el usuario puede acceder a la información de un evento
Precondición	Un usuario está en el catálogo de eventos de una categoría
Acción	El usuario selecciona un evento concreto
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	El usuario ha podido acceder a la información del evento seleccionado

Tabla 184: PNA-10

ID PNA-11

Objetivo	Comprobar que el usuario puede marcar como favorito un evento
Precondición	Un usuario está en el catálogo de eventos de una categoría
Acción	El usuario pulsa en el icono de evento favorito
Resultado esperado	Correcto

Resultado obtenido	Correcto
Postcondición	El usuario ha guardado el evento sobre el que ha pulsado y el icono se ha rellenado de blanco

Tabla 185: PNA-11

ID PNA-12

Objetivo	Comprobar que el usuario puede desmarcar como favorito un evento
Precondición	Un usuario ha marcado como favorito un evento
Acción	El usuario pulsa en el icono de evento favorito
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	El usuario ha desmarcado el evento que había guardado y el icono de favorito vuelve a estar sin rellenar

Tabla 186: PNA-12

ID PNA-13

Objetivo	Comprobar que el usuario tiene acceso a la página web de un evento
Precondición	Un usuario está dentro de la información de un evento concreto
Acción	El usuario accede al hipervínculo de la web del evento
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	El usuario puede entrar en la página web de un evento concreto

Tabla 187: PNA-13

ID PNA-14

Objetivo	Comprobar que los servicios accesibles de cada evento tienen un icono descriptivo
-----------------	---

Precondición	Un usuario está en la información de un evento concreto
Acción	El usuario lee los servicios de accesibilidad que tiene el evento
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	El usuario puede guiarse por los iconos descriptivos que tiene cada servicio accesible

Tabla 188: PNA-14

ID PNA-15

Objetivo	Comprobar que el usuario puede marcar como favorito un evento
Precondición	Un usuario está en el catálogo de eventos de una categoría
Acción	El usuario pulsa en el icono de evento favorito
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	El usuario ha guardado el evento sobre el que ha pulsado

Tabla 189: PNA-15

ID PNA-16

Objetivo	Comprobar que el usuario puede volver a la pantalla principal tocando el logo de la aplicación, situado en la cabecera
Precondición	Un usuario está en el catálogo de eventos de una categoría
Acción	El usuario pulsa en el logo de la aplicación de la cabecera
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	El usuario ha vuelto a la pantalla principal de la aplicación

Tabla 190: PNA-16

ID PNA-17

Objetivo	Comprobar que el usuario puede modificar los datos de su perfil
Precondición	Un usuario está en la pestaña Configuración de la aplicación
Acción	El usuario cambia sus datos personales desde el formulario de Configuración
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	La aplicación permite al usuario modificar su perfil

Tabla 191: PNA-17

ID PNA-18

Objetivo	Comprobar que un usuario invitado no tiene perfil
Precondición	Un usuario invitado está en el menú lateral
Acción	El usuario invitado intenta acceder a su perfil
Resultado esperado	Error
Resultado obtenido	Error
Postcondición	El usuario invitado no encuentra ninguna pestaña de "Mi Perfil"

Tabla 192: PNA-18

ID PNA-19

Objetivo	Comprobar que un usuario invitado no puede acceder a Configuración
Precondición	Un usuario invitado está en la pantalla principal
Acción	El usuario invitado intenta acceder a Configuración
Resultado esperado	Error
Resultado obtenido	Error
Postcondición	El usuario invitado no encuentra ninguna pestaña de "Configuración"

Tabla 193: PNA-19

ID PNA-20

Objetivo	Comprobar que los eventos que se muestran para un usuario invitado son en Madrid con accesibilidad completa
Precondición	Un usuario está en la pantalla principal del sistema
Acción	El usuario invitado entra en el catálogo de eventos de una categoría
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	Los eventos que ve el usuario son de Madrid con accesibilidad completa

Tabla 194: PNA-20

7.1.3 Pruebas de filtro de eventos

ID PFE-00

Objetivo	Comprobar que un usuario puede filtrar por una ciudad y un servicio accesible concreto
Precondición	Un usuario está en la ventana de filtro de eventos
Acción	El usuario selecciona opciones de filtro y los aplica
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	El usuario ha podido filtrar eventos por ciudad y servicio accesible

Tabla 195: PFE-00

ID PFE-01

Objetivo	Comprobar que un usuario no puede filtrar aplicando una opción
Precondición	Un usuario está en la ventana de filtro de eventos
Acción	El usuario selecciona una ciudad concreta y aplica el filtro, sin seleccionar un servicio accesible

Resultado esperado	Error
Resultado obtenido	Error
Postcondición	La aplicación no deja filtrar eventos habiendo seleccionado solo una opción

Tabla 196: PFE-01

ID PFE-02

Objetivo	Comprobar que un usuario puede buscar una ciudad
Precondición	Un usuario está en la ventana de filtro de eventos
Acción	El usuario escribe una ciudad en el formulario de búsqueda de ciudades
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	La aplicación devuelve al usuario la ciudad buscada

Tabla 197: PFE-02

ID PFE-03

Objetivo	Comprobar que un usuario no puede seleccionar una ciudad que no está en el rango de provincias de CESyA
Precondición	Un usuario está en la ventana de filtro de eventos
Acción	El usuario busca una ciudad desconocida
Resultado esperado	Error
Resultado obtenido	Error
Postcondición	La aplicación no devuelve ningún resultado porque esa ciudad no está dentro del rango de provincias definidas

Tabla 198: PFE-03

ID PFE-04

Objetivo	Comprobar que un usuario puede seleccionar una ciudad de forma directa sin tener que buscarla
Precondición	Un usuario está en la ventana de filtro de eventos
Acción	El usuario selecciona una de las ciudades populares
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	El usuario ha podido escoger la ciudad de forma rápida porque la aplicación cuenta con acceso directo de algunas provincias

Tabla 199: PFE-04

ID PFE-05

Objetivo	Comprobar que la aplicación devuelve los eventos de la ciudad y servicio accesible que el usuario filtró
Precondición	Un usuario está en la ventana de filtro de eventos
Acción	El usuario selecciona una ciudad y un servicio accesible y aplica los filtros
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	La aplicación retorna al usuario los eventos de la ciudad y servicio accesible que ha filtrado

Tabla 200: PFE-05

ID PFE-06

Objetivo	Comprobar que la aplicación realiza de forma correcta la petición a los servicios web
Precondición	Un usuario está en la ventana de filtro de eventos
Acción	El usuario selecciona una ciudad y un servicio accesible y aplica los filtros
Resultado esperado	Correcto
Resultado obtenido	Correcto

Postcondición

La aplicación retorna al usuario los eventos de la ciudad y servicio accesible que ha filtrado

Tabla 201: PFE-06

ID**PFE-07**

Objetivo	Comprobar que el buscador de ciudades reconoce acepciones incorrectas, en otro idioma o no formales del nombre de las provincias
Precondición	Un usuario está en la ventana de filtro de eventos
Acción	El usuario busca la ciudad “Islas Baleares”
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	La aplicación retorna al usuario la ciudad “Illes Balerars”

Tabla 202: PFE-07

ID**PFE-08**

Objetivo	Comprobar que un usuario invitado no puede filtrar eventos
Precondición	Un usuario invitado está en el catálogo de eventos de una categoría
Acción	El usuario invitado intenta filtrar eventos
Resultado esperado	Error
Resultado obtenido	Error
Postcondición	El usuario invitado no encuentra ninguna pestaña para filtrar los eventos

Tabla 203: PFE-08

7.1.4 Pruebas de almacenamiento de datos

ID**PAD-00**

Objetivo	Comprobar la correcta conexión con la base de datos
-----------------	---

Precondición	Un usuario ha abierto a la aplicación
Acción	El usuario se registra en la aplicación desde el formulario de registro
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	El usuario se ha registrado, almacenándose su información en la base de datos

Tabla 204: PAD-00

ID PAD-01

Objetivo	Comprobar que se guardan los eventos que un usuario marca como favorito
Precondición	Un usuario está en el catálogo de eventos de una categoría
Acción	El usuario marca como favorito un evento
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	El evento que ha marcado el usuario se ha almacenado en la base de datos, dentro de la colección <i>Users</i>

Tabla 205: PAD-01

ID PAD-02

Objetivo	Comprobar la consulta de información de la base de datos
Precondición	Un usuario se ha logueado en la aplicación
Acción	El usuario accede a su perfil
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	El perfil del usuario muestra su información personal, consultada en la base de datos

Tabla 206: PAD-02

ID PAD-03

Objetivo	Comprobar la correcta lectura de la información de los servicios web
Precondición	Un usuario está en el catálogo de eventos de una categoría
Acción	El usuario accede a la información de un evento
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	La aplicación devuelve toda la información del evento, extraída de los servicios web de CESyA

Tabla 207: PAD-03

ID PAD-04

Objetivo	Comprobar que se realizan cambios en la base de datos cuando un usuario modifica su perfil
Precondición	Un usuario en Configuración
Acción	El usuario modifica sus datos personales y guarda los cambios
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	La base de datos ha actualizado los campos que el usuario ha cambiado

Tabla 208: PAD-04

ID PAD-05

Objetivo	Comprobar que un usuario invitado no puede guardar información en la base de datos
Precondición	Un usuario invitado está en el catálogo de eventos de una categoría
Acción	El usuario invitado pulsa el icono de evento favorito de uno concreto
Resultado esperado	Error

Resultado obtenido	Error
Postcondición	El usuario invitado no puede marcar un evento como favorito, mostrándose que no puede almacenar información en la base de datos del sistema

Tabla 209: PAD-05

ID PAD-06

Objetivo	Comprobar el funcionamiento del sistema cuando la base de datos está caída
Precondición	Un usuario ha entrado en la aplicación
Acción	Un usuario se va a registrar en el sistema
Resultado esperado	Correcto
Resultado obtenido	Correcto
Postcondición	El sistema pone en cola la petición de registro de este usuario para que se lleve a cabo cuando Firebase vuelva a funcionar o a tener conexión

Tabla 210: PAD-06

8. IMPLANTACIÓN DEL SISTEMA

Cuando el proyecto se ha desarrollado y se ha comprobado su validez mediante un plan de pruebas, el siguiente paso es su implantación. Al tratarse de un TFG, este proyecto no se ha llegado a subir las plataformas de compra de aplicaciones, pero se va a detallar cómo sería su implantación. Estas plataformas serían:

- **Play Store** para dispositivos *Android*
- **App Store** para dispositivos *IOS*

Cada dispositivo en el que se descargue la aplicación ACA, ya sea desde Play Store o desde App Store, gestionaría la aplicación por los servidores propios de cada plataforma de compra, quien desplegaría el sistema. La base de datos que gestione el sistema seguirá siendo la misma de este proyecto, al igual que la conexión a los servicios web de CESyA.

Aunque este proyecto no se pueda descargar de forma real, sí se puede comprobar su implementación en simuladores de cada plataforma:

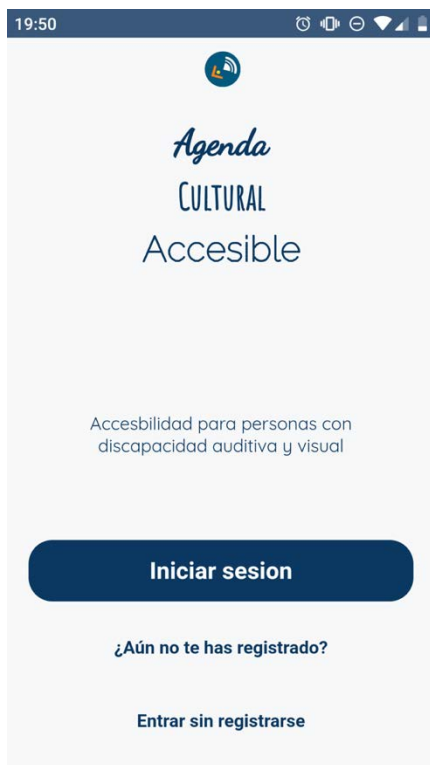


Ilustración 51: Implantación en Android

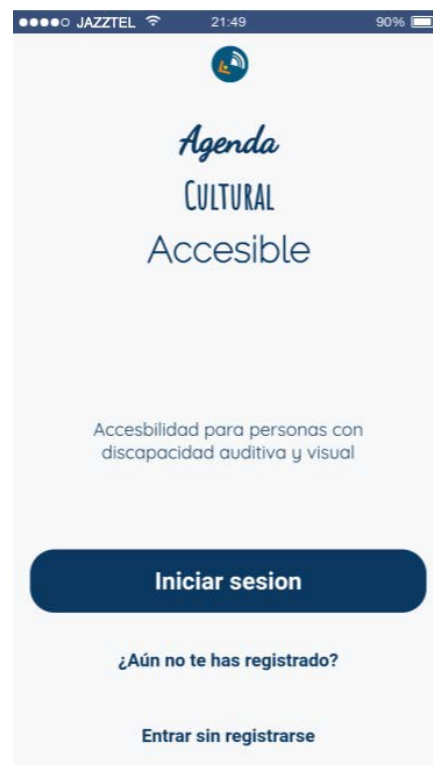


Ilustración 52: Implantación en IOS

Para los dispositivos *Android*, se han utilizado tanto simuladores del entorno de ejecución Android Studio como un propio dispositivo móvil con *Android*. Para los dispositivos *IOS*, se ha utilizado el simulador de XCode y un dispositivo móvil con *IOS*.

9. CONCLUSIONES

Una vez finalizado todo este proyecto, es un placer comentar las conclusiones que se han sacado.

Para empezar, es un punto muy positivo que los alumnos tengan que enfrentarse de forma individual a este tipo de trabajos. Aporta una mayor perspectiva de la realidad, alejándose de lo académico que puede ser un trabajo de la universidad. Proporciona una responsabilidad mayor y necesaria para el mundo laboral, porque se aprende a enfrentarse a problemas reales, y a solventarlos, o al menos, a intentarlo.

Por otra parte, este proyecto contó con alguna barrera de entrada, como por ejemplo el desconocimiento del entorno de desarrollo en el que se ha definido todo el sistema. Se han necesitado varios meses para afianzarse a esta herramienta. Bien es cierto que no se trata de un entorno complicado, pero es totalmente diferente a todo lo visto en el grado, por lo que se partía desde cero. No obstante, esto ha implicado un nuevo conocimiento y por tanto se ha ampliado el abanico de opciones a elegir cuando se desee definir algún otro sistema.

Otra barrera de entrada fue el desconocimiento total de la de creación de aplicaciones para dispositivos móviles. Es un ámbito muy interesante pero nunca se había tocado. Sí se conocía el desarrollo web, pero no para móviles.

El entorno de desarrollo escogido ha exigido que no se salga de la programación de estos tres lenguajes en concreto: HTML, JavaScript y CSS. El alumno contaba con conocimientos avanzados sobre estos lenguajes de programación, ya que se han visto en diversas asignaturas. Sin embargo, aunque ha sido de ayuda tener esta base previa, se han aprendido funciones y formas de desarrollar nuevas, sobre todo en JavaScript. Por ejemplo, nunca antes se había tratado peticiones a servicios web con este lenguaje. Esto ha ayudado reforzar los conocimientos acerca de cómo funciona un servicio web, cómo se lee y extrae información de ellos, etc. Esto es otra de las cosas que ha aportado conocimientos nuevos al alumno.

Para todo este proyecto se han podido aplicar muchos conocimientos que se han aprendido en la carrera. En general, lo que más se ha tocado ha sido lo aprendido en asignaturas relacionadas con la ingeniería de software, como *Principios de Desarrollo Software (PDS)*, *Dirección de Proyectos de Desarrollo Software (DPDS)* y *Ingeniería de Software (IS)*, entre otros. Sobre todo, se agradece haber cursado asignaturas como DPDS porque muestra la realización completa de del desarrollo de un producto software. Gracias a lo aprendido en ella, este proyecto se ha podido realizar de una forma más cómoda, porque ha sido el desarrollo de un proyecto software de principio a fin, como se vio en la asignatura.

También se ha podido poner en práctica todo lo aprendido en asignaturas como *Tecnologías Informáticas para la Web* o *Interfaces de Usuario*. Por otro lado, aunque durante el grado se considerasen de las más complejas, todas aquellas asignaturas en las que se trata tanto la administración como el diseño de bases de datos ha sido muy útil para este proyecto, por haber aportado una imagen más concreta de la gestión de los datos. Gracias a ellas, el uso de bases de datos en este proyecto no solo no ha sido un problema, sino que ha sido muy grato saber manejarlas.

Gracias a la realización de este proyecto, se ha podido ver la importancia del análisis y el diseño de un sistema de información. En todo desarrollo software puede existir la idea de implementar el producto sin necesidad de analizar cómo hacerlo ni cómo diseñarlo antes de implementarlo. Al menos en caso del alumno, era algo muy abstracto de entender. Habiendo realizado este proyecto se ha comprobado en primera persona la necesidad de analizar y diseñar todo un sistema antes de construirlo.

Es muy grato haber aprendido a desarrollar aplicaciones móviles, y además, haber aprendido un nuevo paradigma de definición de aplicaciones móviles, como es el desarrollo multiplataforma. Esto ha hecho ver al desarrollador de ACA su gusto por el desarrollo tanto web como móvil. Esto es un punto fundamental, ya que ayuda al desarrollo profesional del alumno, por haber visto en qué rama de todo lo aprendido quiere desempeñar su carrera laboral.

Se espera que esta aplicación, en un futuro, pueda ayudar, en un ámbito real, a todas las personas que necesiten de algún tipo de accesibilidad.

9.1 Posibles mejoras del sistema

En este apartado se van a describir las posibles mejoras que podrían aplicarse al sistema para mejorar su rendimiento, usabilidad y funcionalidad.

- Permitir al usuario darse de baja de los sistemas de ACA: ahora mismo la aplicación no cuenta con la posibilidad de dar de baja al usuario en caso de que lo desee. Esta función aportaría mayor libertad al usuario.
- Valoración de los eventos: añadir puntuación en formato de estrella sobre los eventos, para que sirva de *feedback* tanto para los usuarios, como para los creadores del evento, pudiendo aplicar mejoras.
- Aplicar al desarrollo del sistema los reglamentos y directivas de una aplicación accesible: actualmente ACA ofrece servicios accesibles, pero no sigue un desarrollo accesible. Crear un desarrollo que cumpla todos los términos de accesibilidad.
- Añadir audio a la aplicación: incluir notas de voz sobre la aplicación que sirvan de guía en la navegación, para quien no pueda ver la aplicación.
- Mejorar la estructura de la base de datos: añadir nuevas funcionalidades y colecciones sobre la base de datos.
- Mejorar la usabilidad de la aplicación: seguir añadiendo funcionalidades que hagan de ACA una aplicación 100% usable por todo tipo de usuario.

10. BIBLIOGRAFÍA

1. Sistemas Operativos Móviles .

<https://www.areatecnologia.com/informatica/sistemas-operativos-moviles.html>.

2. Computer Hoy. *La guerra de los smartphones*.

<https://computerhoy.com/reportajes/industria/android-vs-iphone-guerra-smartphones-cifras-271447>.

3. Líderes del mercado: Android vs IOS.

<https://computerhoy.com/reportajes/industria/android-vs-iphone-guerra-smartphones-cifras-271447>.

4. 480. *¿App híbrida o app nativa?*

<https://cuatroochenta.com/app-hibrida-o-app-nativa-segun-para-que/>.

5. Administración Electrónica . *MÉTRICA*.

https://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Metrica_v3.html#.XG3DxS1DnjA.

6. OBS Business School. *¿Qué es un diagrama de Gantt y para qué sirve?*

<https://www.obs-edu.com/es/blog-project-management/diagramas-de-gantt/que-es-un-diagrama-de-gantt-y-para-que-sirve>.

7. CESyA. *¿Quiénes somos?*

<http://www.cesya.es/quienes>.

8. Portal Administración Electrónica. *Métrica*

[v3.https://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Metrica_v3.html#.XG3DxS1DnjA](https://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Metrica_v3.html#.XG3DxS1DnjA).

9. Reglamento General de Protección de Datos .

Nuevo reglamento. <https://rgpd.es>.

10. BOE - Agenda Estatal Boletín Oficial del Estado. *Documento consolidado BOE-A-2018-16673*.

<https://www.boe.es/buscar/act.php?id=BOE-A-2018-16673>.

11. Apache Cordova. *Resumen*.

<https://cordova.apache.org/docs/es/latest/guide/overview/>.

12. 480. *¿App híbrida o app nativa?*

<https://cuatroochenta.com/app-hibrida-o-app-nativa-segun-para-que/>.

13. Samir Kamble. *Apache Cordova Architecture*.

<https://www.samirkamble.com/installing-and-setting-up-apache-cordova-phonegap-environment/>.

14. Ok Diario. *¿Qué es Xamarin?*

<https://okdiario.com/tecnologia/2018/03/25/que-xamarin-2022974>.

15. Slide Share. *Xamarin*.

<https://www.slideshare.net/pauloortins/use-the-xamarinforms-and-surprise-your-customers-when-develop-native-apps-in-less-time-and-cheaper>.

16. Skelia. *Xamarin VS Apache: Which one is better?*

<https://skelia.com/articles/xamarin-vs-apache-cordova-one-better/>.

17. Firebase.

<https://firebase.google.com>.

18. Braintive. *10 reglas heurísticas de usabilidad de Jakob Nielsen*.

<http://www.braintive.com/10-reglas-heuristicas-de-usabilidad-de-jakob-nielsen/>.

19. ECMA International. *EcmaScript language Specification*.

<https://www.ecma-international.org/ecma-262/6.0/>.

ANEXO I: MANUAL DE USO

A continuación, se va a mostrar el manual de uso de la aplicación para el usuario final. Este manual expresa cómo ha de usarse la aplicación correctamente.

1. Manual de usuario

1.1 Acceso a la aplicación

La aplicación ACA puede descargarse tanto para *Android* como para *IOS*. Por tanto, está disponible en las tiendas de aplicaciones, de carácter gratuito:

- *Play Store (Android)*
- *App Store (IOS)*

ACA permite diferentes formas de acceso:

1. Crearse una cuenta registrándose en el sistema y acceder con ella a la aplicación
2. No crearse una cuenta y acceder directamente a la aplicación

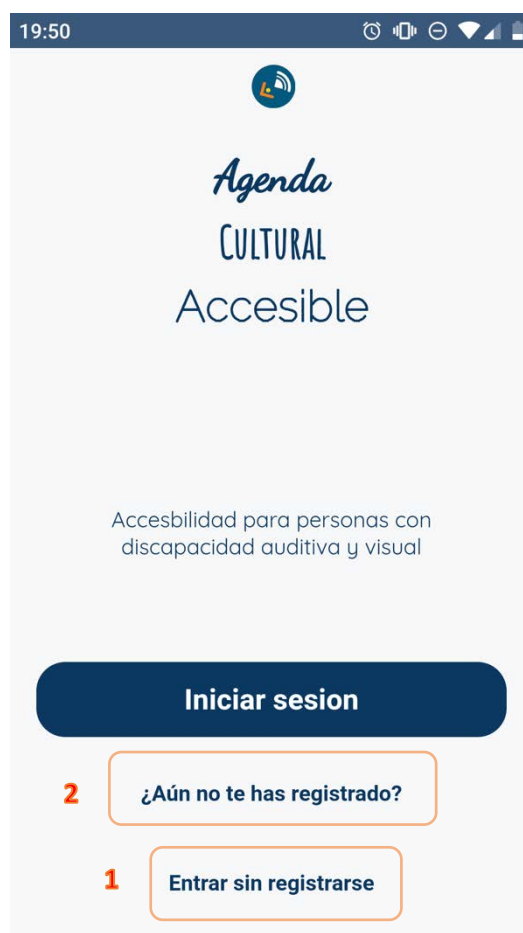


Ilustración 53: Manual Usuario - Pantalla inicial

1.2 Registrarse en la aplicación

Para crearse una cuenta es necesario completar el formulario de registro:



Ilustración 54: Manual Usuario - Registro I

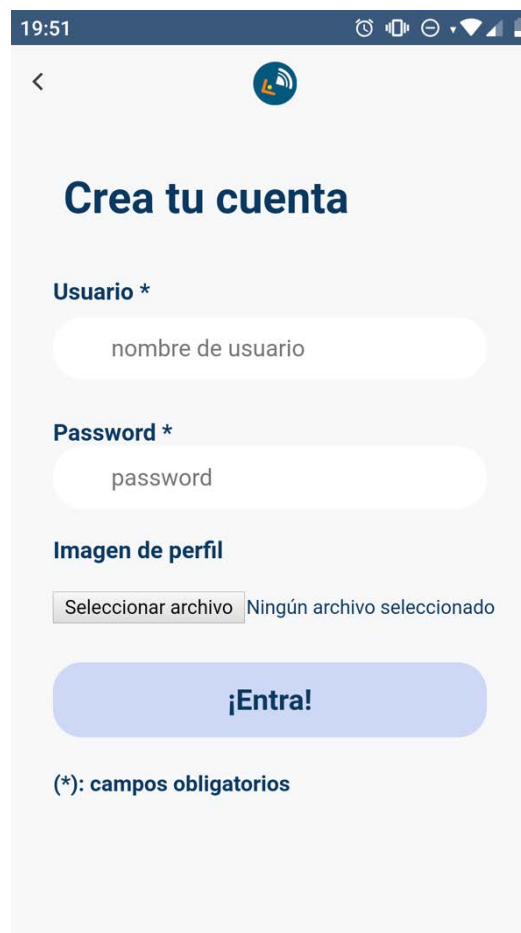
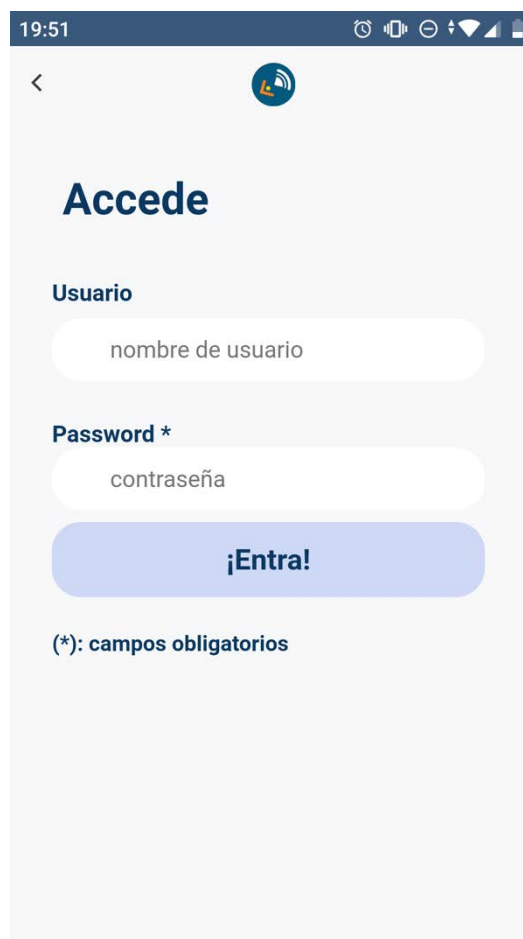


Ilustración 55: Manual Usuario – Registro II


La aplicación necesita el nombre completo del usuario, una cuenta de correo electrónico, un nombre de usuario y una contraseña. Este nombre de usuario debe ser único, es decir, no puede haber ya otro usuario registrado con ese mismo nombre. Además, se puede añadir una foto de perfil aportando más información a la cuenta.

1.3 Iniciar sesión en la aplicación

Una vez se tiene la cuenta creada, el paso siguiente será entrar en la aplicación con ella:



19:51

< 

Accede

Usuario

nombre de usuario

Password *

contraseña

¡Entra!

(*) : campos obligatorios

Ilustración 56: Manual Usuario - Inicio de sesión

El formulario de inicio de sesión requiere que se complete con el nombre de usuario y la contraseña. Si alguno de ellos no es correcto, no permitirá al usuario entrar en ACA. Por el contrario, cuando estos datos sean correctos se accederá a la aplicación.

1.4 Pantalla principal de la aplicación

La pantalla principal tiene diferentes funciones. En primer lugar, se puede acceder a las diferentes categorías que tiene la aplicación: museo, teatro, cine y otros eventos. También es posible acceder a *Mi Archivo*, el almacén donde están los eventos que se han marcado como 'favorito'. Otra tercera función que tiene esta pantalla es el acceso a *CESyA*, que proporciona información sobre la organización que lleva los eventos de esta aplicación.

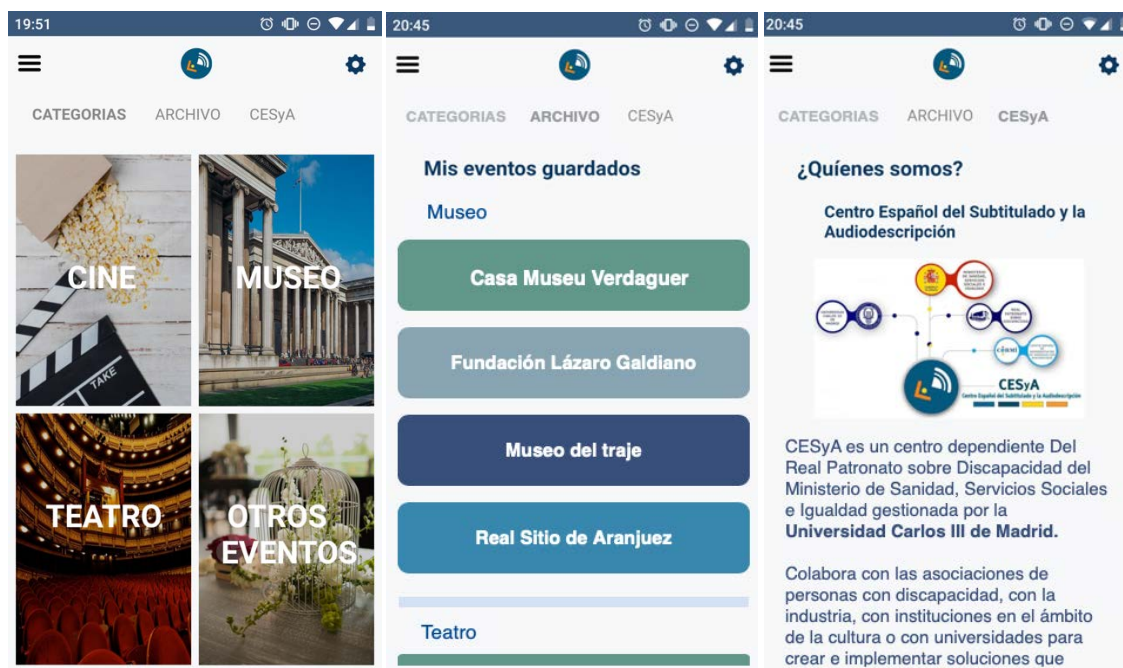


Ilustración 57: Manual de Usuario – Opciones pantalla principal

1.5 Selección y filtro de eventos

Para acceder a los eventos accesibles es necesario escoger un género del catálogo de categorías.



Ilustración 58: Manual de usuario - Eventos

Para este manual se va a tomar de ejemplo el género “Museo”. Se tiene acceso a todos los eventos disponibles para este género. Por defecto, se muestran los eventos accesibles

disponibles en Madrid con accesibilidad completo. No obstante, el usuario puede cambiar esta configuración y aplicar los filtros que el desee:

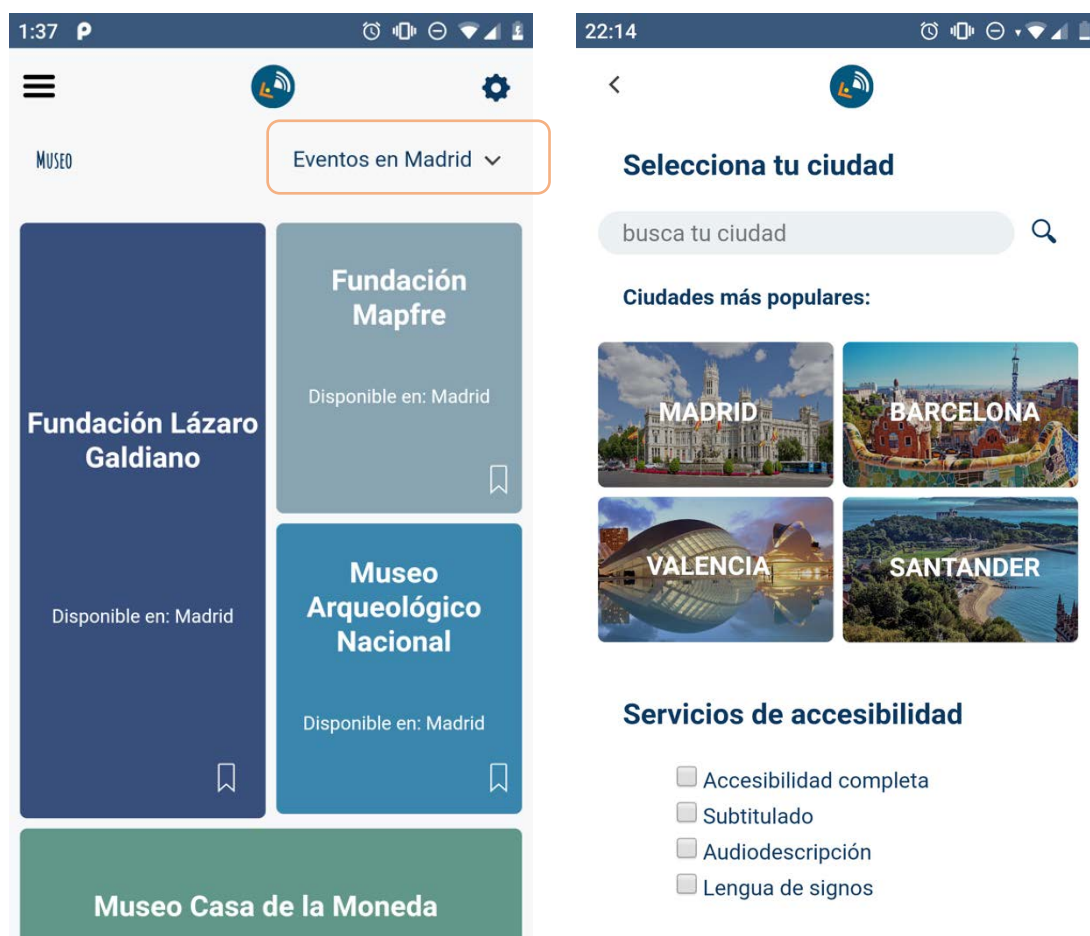


Ilustración 59: Manual Usuario - Filtro de eventos

Para filtrar eventos, el usuario debe entrar en “Eventos en Madrid”. Se ha de escoger estas opciones:

- **Ciudad** en la que se desarrollen los eventos accesibles
- **Tipo de accesibilidad** que definan a los eventos

Para filtrar por ciudad, la aplicación pone a disposición del usuario unas cuantas ciudades de manera rápida, que son las más populares del país. Si la ciudad que el usuario desea buscar no está entre ellas, puede buscarla. Estas son las ciudades disponibles en ACA:

- Álava, Albacete, Alicante, Almería, Asturias, Ávila, Badajoz, Barcelona, Burgos, Cáceres, Cádiz, Santander, Castellón, Ceuta, Ciudad Real, Córdoba, A Coruña, Cuenca, Girona, Granada, Guadalajara, Guipúzcoa, Huelva, Huesca, Illes Balears, Jaén, León, Lleida, Lugo, Madrid, Málaga, Melilla, Murcia, Navarra, Ourense, Palencia, Las Palmas, Pontevedra, La Rioja, Salamanca, Santa Cruz de Tenerife, Segovia, Sevilla, Soria, Tarragona, Teruel, Toledo, Valencia, Valladolid, Vizcaya, Zamora y Zaragoza.

El nombre de la provincia se debe escribir correctamente, sin necesidad de escribir tildes. Aún así, la aplicación, para evitar errores de escritura, admite formas semánticas incorrectas de la ciudad, a causa de que el usuario lo escriba en otro idioma o que lexicalmente esté mal escrita. Por ejemplo, si el usuario introduce “Baleares”, ACA reconocerá esta ciudad, o si escribe “Guipuzcua” (en vez de “Guipúzcoa”) también esta ciudad será reconocida.

Cuando se ha seleccionado la ciudad a filtrar, el siguiente paso es seleccionar el tipo de accesibilidad. Se ha de escoger una opción de todas las que hay:

- a) Accesibilidad completa
- b) Subtitulado
- c) Audiodescripción
- d) Lengua de signos

Para poder aplicar estos filtros sobre los eventos, se tienen que seleccionar los dos. El usuario no puede filtrar eventos solo por ciudad o solo por servicio accesible.

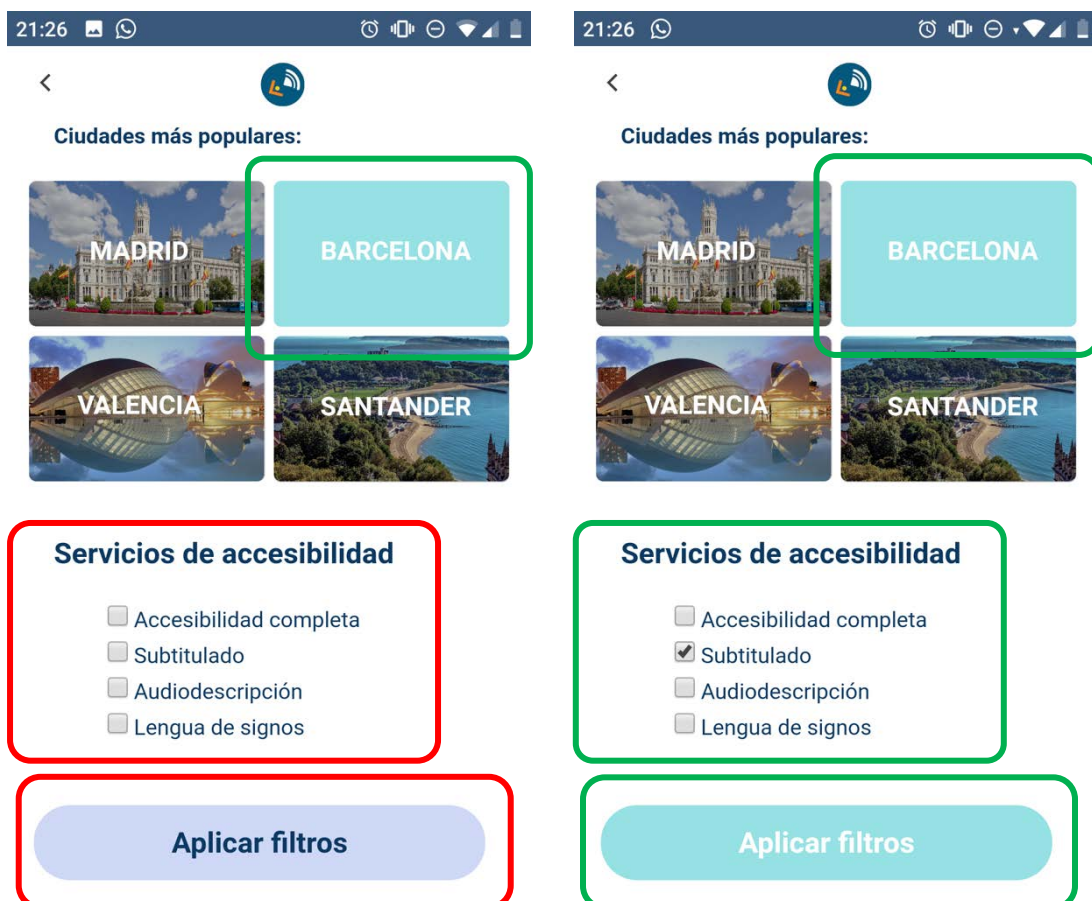


Ilustración 60: Manual Usuario - Filtro de eventos

Para ver la información de un evento basta con seleccionar cualquiera de ellos:



Ilustración 61: Manual Usuario - Información de un evento

Desde esta ventana, el usuario puede acceder a las páginas web de información adicional del evento.

1.6 Marcar favorito un evento

Un usuario puede marcar como favorito un evento. Para ello, tiene que tocar el icono en forma de aspa que tiene cada evento, que quedará relleno si se ha marcado:

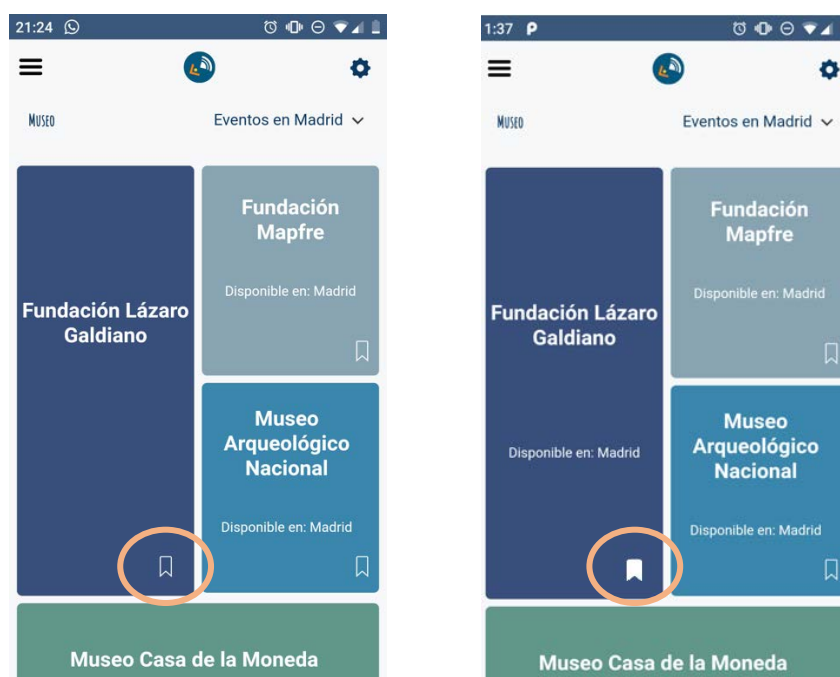


Ilustración 62: Manual Usuario - Marcar favorito

1.7 Desmarcar evento como favorito

Para desmarcar un evento como favorito, bastará con volver a tocar el icono en forma de aspa, quedando éste sin rellenar.

1.7 Volver a pantalla principal

Para volver a la pantalla principal se puede tocar en el logo de ACA, que está situado en la parte superior:



Ilustración 63: Manual Usuario - Volver a pantalla principal

1.8 Acceder y modificar el perfil del usuario

1.8.1 Acceder al perfil

Para acceder al perfil del usuario, se puede acceder desde el menú lateral:

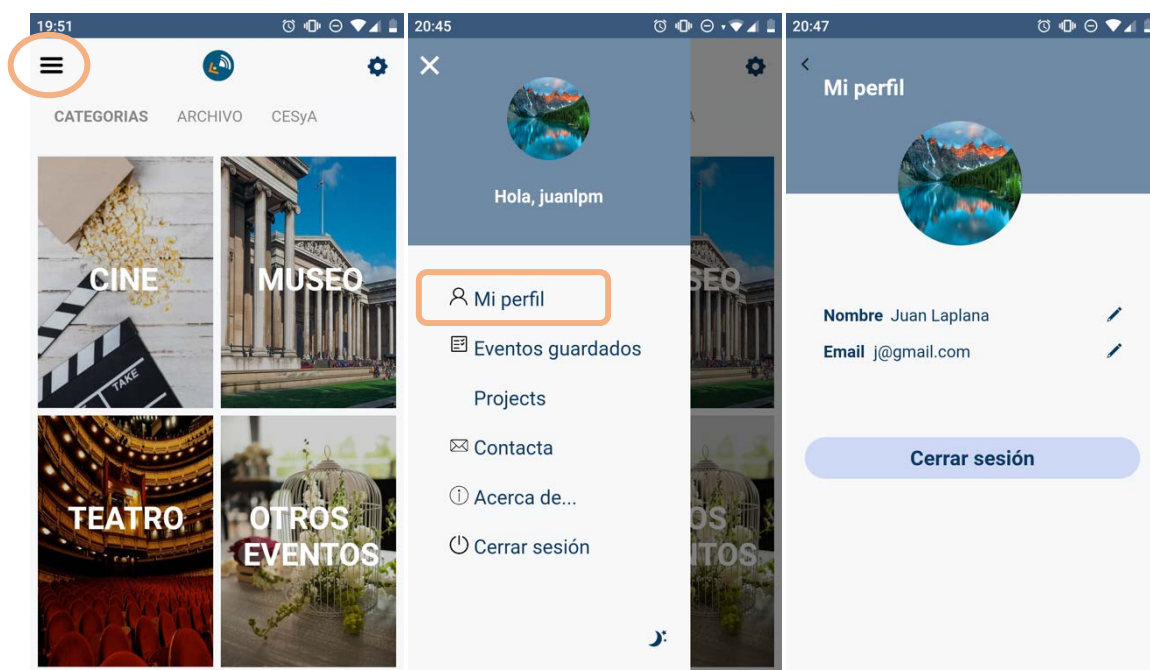


Ilustración 64: Manual Usuario - Acceso a Mi Perfil

1.8.2 Modificar perfil

Desde el perfil del usuario se pueden modificar los datos de la cuenta, accediendo desde el icono en forma de lápiz. En la pantalla Configuración es donde se cambia esta información:

 The image shows two screenshots of the 'Edita tu perfil' (Edit your profile) form.
 - The first screenshot (20:45) shows the form fields: 'Nombre y apellidos *' (Juan Laplana), 'Email *' (j@gmail.com), 'Usuario *' (juanlpm), 'Nueva password *', and 'Repite la password *'.
 - The second screenshot (20:45) shows the 'Imagen de perfil' section with a 'Seleccionar archivo' button and the text 'Ningún archivo seleccionado'. Below this is a 'Guardar cambios' button and a note '(*) : campos obligatorios'.

Ilustración 65: Manual Usuario - Editar datos del perfil

Dentro de este formulario, el usuario puede cambiar desde todos los datos, hasta ninguno. Por ejemplo, puede cambiar su cuenta de correo electrónico sin tener que

cambiar el resto de los campos. O puede cambiar su contraseña manteniendo el resto de su información igual.

Para acceder a este formulario también se puede desde la pantalla principal, o desde cualquier ventana que tenga el icono de “configuración”:

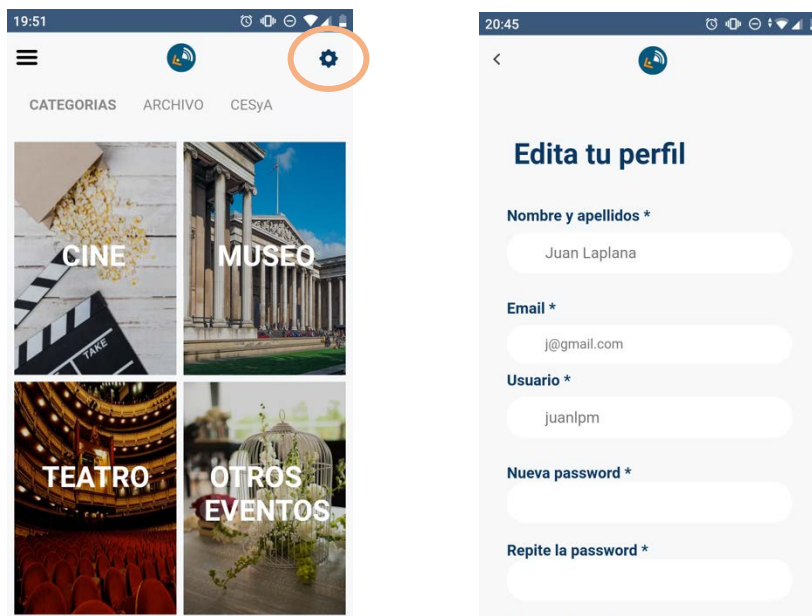
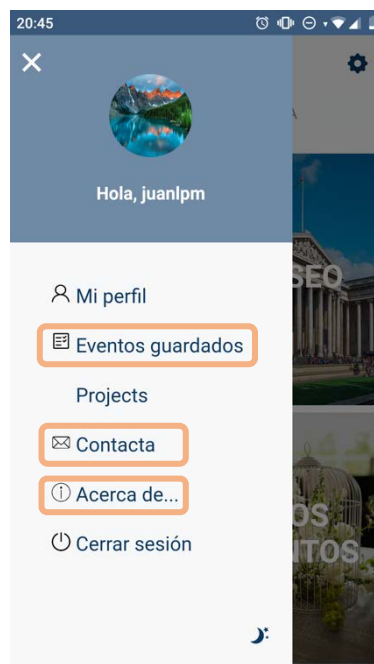


Ilustración 66: Manual Usuario - Acceso a Configuración

1.9 Información sobre CESyA

El usuario podrá contactar con CESyA, o informarse sobre ella. Esto se realiza desde el menú lateral, donde también se puede acceder a los eventos guardados:



1.10 Cerrar sesión

Finalmente, el usuario podrá cerrar sesión tanto desde su perfil como desde el menú lateral.

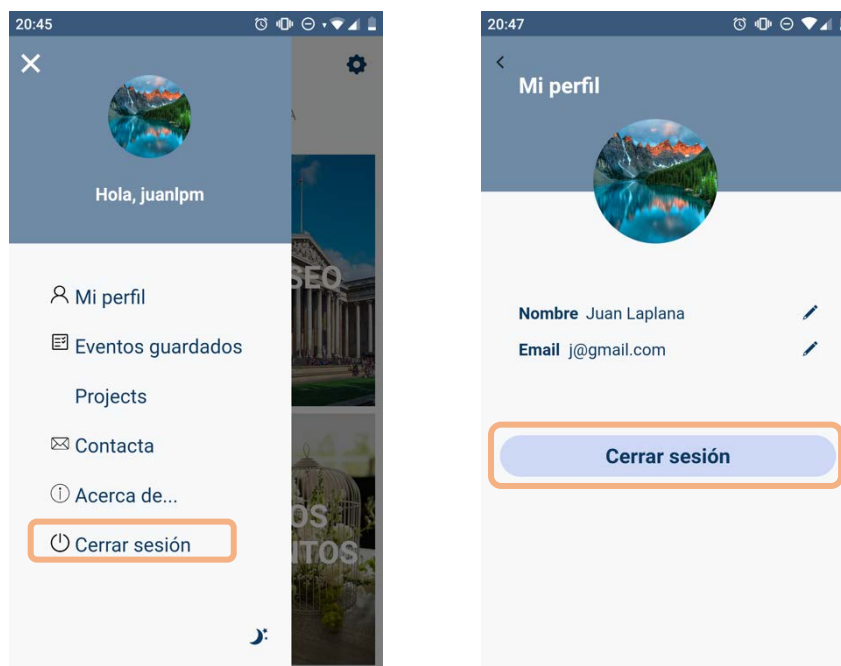


Ilustración 67: Manual Usuario - Cerrar sesión

1.11 Entrar en la aplicación sin registrarse

Para entrar en la aplicación sin registrarse, simplemente se ha de escoger la opción 2 de acceso: “¿Aún no te has registrado?”. Esta opción permitirá entrar al usuario sin necesidad de guardar sus datos en el sistema. Sin embargo, no podrá disfrutar de todos los servicios que ACA ofrece. Estará en modo invitado.

1.11.1 Acceder a pantalla principal sin haberse registrado

Desde la pantalla principal se puede acceder a los diferentes géneros que ACA tiene de eventos accesibles. También se podrá ver Mi Archivo, el almacén donde se guardarían los eventos que el usuario desee, si estuviese registrado. De igual forma, se puede acceder a CESyA, una ventana donde se informa sobre la organización que mantiene estos eventos.

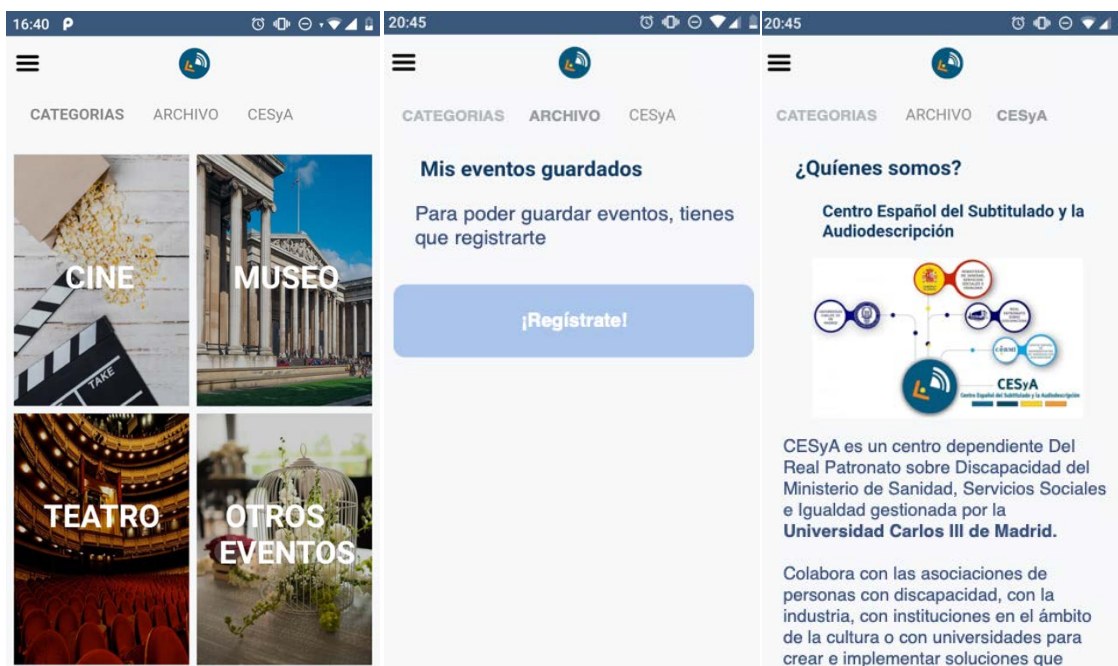
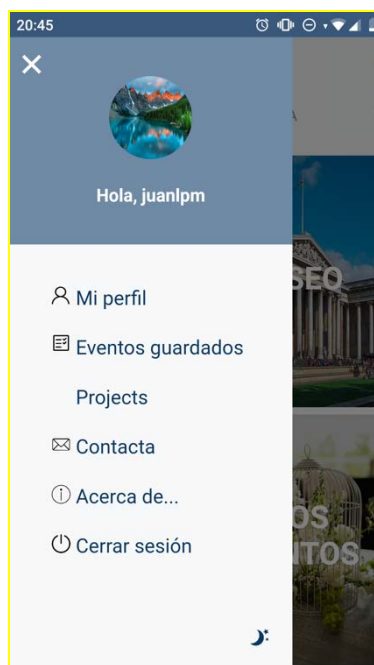


Ilustración 68: Manual Usuario - pantalla principal usuario invitado

1.11.2 Acceder a información de la aplicación

Para acceder a información de la aplicación, el usuario invitado puede desplegar el menú lateral:



Desde este menú tendrá opción a registrarse siempre que lo desee. También puede contactar e informarse de CESyA, la organización que gestiona los eventos de ACA.

1.11.3 Seleccionar eventos sin haberse registrado

El usuario invitado puede seleccionar cualquier evento de los que ACA ofrece, pudiendo ver toda su información. Por defecto, se muestran los eventos accesibles disponibles en Madrid con accesibilidad completa. Si el usuario invitado quisiera ver qué eventos hay en otra ciudad o con otro servicio accesible, tendría que registrarse.



Ilustración 69: Manual Usuario - Eventos de usuario invitado

Desde la ventana de la información de cada evento el usuario puede acceder a las paginas web con información adicional del evento.

1.11.4 Volver a pantalla principal

El usuario invitado podrá volver a la pantalla principal desde el logo de ACA, situado en la parte superior:



Ilustración 70: Manual Usuario - Volver a pantalla principal usuario invitad

ABSTRACT

1. Introduction

The purpose of this document is to present the development of an end-of-studies work of degree in computer engineering. It is an academic work where many of the knowledge learned during the degree are put into practice.

The work carried out consists on the definition of a multiplatform application on mobile devices for accessible events.

1.1 Current social context

Why a cross-platform application? This is one of the questions that arise when you start a mobile application development.

Today, the market for mobile brands is very divided, and not because of the aesthetic design that can offer a specific brand, but for the operating system that includes the mobile. There are more than 8 different operating systems for mobile devices, such as Android, IOS, Windows Phone, Blackberry Symbian, etc. Well it is true that at present the most predominant are *Android*, own Google and *Ios*, Of Apple.

The battle between Android and IOS is something that takes a long time to go, and will not end. So, following the information provided by the preceding paragraph, the rest of Systems Operating as competition in the market, leaving these two leaders.

Analyzing society, there are many people who prefer Android on iOS, and another percentage of people who prefer iOS on Android. Currently, this is the percentage of users using each platform:

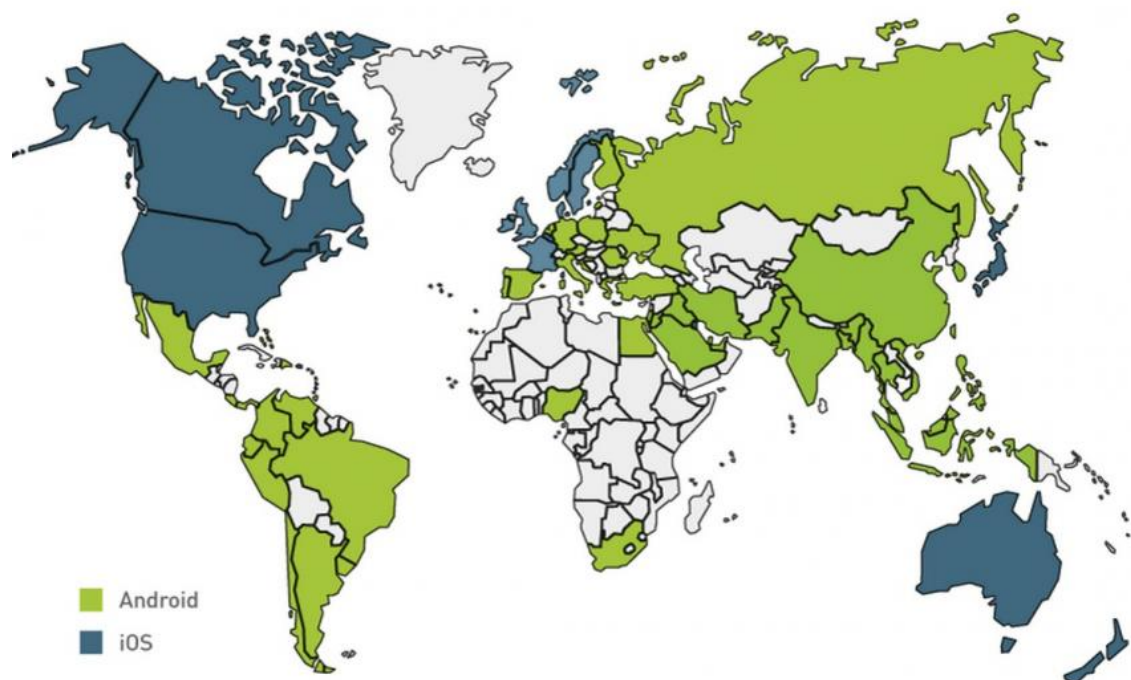


Illustration 71: Android and IOS Domain

With illustration 1 It is clear that Android is much more used by users than IOS. This has a clear explanation and is that, away from the preferences of each user, Android is an operating system that is implemented in most Brands Mobile market. That's why What, by probability, there are more mobiles with this system.

This incessant struggle between these operating systems leads to the question of which platform to develop a mobile application. Every good developer could create a native project for each operating system, and thus expand the range of sales possibilities in the market. However, why not create an application where a single software project serves all the platforms you want? That's why What There is multiplatform development, which, in addition to covering this minority problem, brings many advantages to mobile development.

Knowing this analysis of the current situation of the development of applications, the idea of defining an application for accessible events goes by the hand that this one is multiplatform. In addition, the platforms in which it would be appropriate to develop the application are the market leaders, Android and IOS.

1.2 Motivation

The main motivation to create an application of accessible events has been the tutor of this project. Currently, there are applications that provide events with accessibility services, but do not have a native development.

That is why, knowing the context of the definition of mobile applications, has wanted to bet on the creation of a hybrid development. The main advantages that will be covered are the improvement in the maintenance, performance and scalability of a project.

Another advantage that has been chosen this development model is because it is oriented to the distribution of content and the application that you want to define distributes the content of what events are available in different places.

In addition, another reason for this project is learning and familiarization with this new paradigm of creation of applications, since it is in full growth.

This project will encompass users of both Android and IOS, without having to run the risk of not being able to go to the market on any of these platforms. If this development had been native, still being able to create several native systems of the same application, perhaps one would have chosen a single platform, by the simplicity and simplicity in when to development and maintenance of the code.

1.3 Project Objectives

The objectives to cover with this project are the following:

- Learn how to develop applications for mobile devices
- Learn the paradigm of multiplatform application creation
- Covering the two market-leading mobile operating systems
- Apply the maximum knowledge of a software development learned in the degree

- Create an application that, by providing accessible events, can be developed and maintained in an optimal way
- Create a simple and easy-to-use application for users with accessibility needs
- Show the available events in a given city with a specific accessibility type
- Allow access to the application registering and unregistered, but by applying restrictions on users who do not register
- Store personal information from users using the application, under Data protection
- Get event information through requests to Web services

1.4 Document Structure

This document consists of the following parts:

- **Introduction:** Provides a global vision of the project, showing the main ideas that have moved this development and analyzing the current context. The objectives that this project has to achieve are also studied. Includes the description of the parts that make up this document.
- **Definition and Acronyms:** This section includes the explanation of terms that may be technical and unknown by the reader and the acronyms and/or abbreviations that have been used throughout the document.
- **Project Management:** Specifies the details of each task to be performed in this project, counting the lifecycle, organization and planning of the same, to achieve a quality software PRODUCT.
- **Feasibility study of the system:** In this section we study the context in which the development of this system, comparing it with competing applications and analyzing possible alternatives of development of The same. In this section, the option to carry out the project will be chosen.
- **System Analysis:** An analysis will be carried out prior to the development of the application, studying its definition, the user's needs to create this project, the data models that will be followed and the user interfaces of the application will be designed. All this is validated with traceability matrices to check the consistency of the system.
- **System Design:** This section builds what was previously analyzed for the creation of this project. The architecture of the system is defined, the interaction between each element of its structure and the physical model that will follow the database is designed.
- **System Validation:** A test plan will be created to validate that the software product acts as expected.
- **System implementation:** This section details the implementation of the system, once developed and validated.
- **Conclusions:** This section explains the conclusions that the student has taken from the development of this project.
- **Bibliography:** This section lists all the Web pages or bookstores that the student has used and in which he has supported the development of this software project.
- **User Manual:** This document contains a user manual so they know and can use it, indicating step-by-step actions that allow the application, preventing it from being in error situations.
- **Abstract:** Part written in English where the introduction and conclusions of the project are narrated

2. Feasibility study of the system

This task will study how viable is the creation of this system, analyze the context in which it will develop, seeing the possible problems that the current situation. Different alternatives to the solution will be analyzed, based on what the system needs.

2.1 Setting the scope of the system

2.1.1 Study of the application

The purpose of this project is to capture different types of events accessible to people with auditory and visual impairment in an application. In this way, it is exposed in a way closer to the user this type of events for all the people that need or require these accessibility.

The application you want to develop is called ACA. It is a cultural agenda where you can consult the events that are available in certain cities, and with a specific accessibility service. The genre of these events is very varied, as it can be theatre, museum, cinema or other without specific classification. The accessibility services offered are subtitles, audiodescription, sign language or full accessibility, which includes the last three.

The user can consult these events in different cities of Spain, all those that enter within the framework of cities that CESyA offers.

Another purpose of this application is to have the application cross-platform, covering the platforms *Android* e *ios*. It is intended to obtain a development that is worth for both operating systems, performing these aspects:

- Develop a single code for the system
- Approximate the development and maintenance of the application to a single module
- Improve the quality and scalability that a native system would offer

2.1.2 Identification of the scope of the system

This section relates the capacity of the application to solve the needs for which it has been raised. ACA has two types of users:

- **Registered user:** This type of user has registered in the application creating an account from a form. You have access to all the services offered here.
- **Guest user:** This type of user does not need to create an account. You can access the application, but it has limited access to the services offered here.

To access ACA, a user is in the initial window three options: Login, register or enter without registering. Previously identified the different roles that the Application and in view of the initial window, different scopes of the system are differentiated, depending on the type of user:

2.1.2.1 scope of system: Registered user

A registered user is the one who creates an account to access to in the application. Therefore, in the initial window accesses the option "Have you not registered yet?" To access register here. This section has a form, which is divided into two windows. This form has these fields:

- Name and surname (obligatory)
- Email (required)
- User name (required)
- Password (required)
- Profile image (optional)

To carry out the registration, there can be no email in the ACA database, or another user name equal to yours. Filling in these fields and following the necessary requirements, the user will be able to finally create his account and save his data in the databases of ACA, to be able to access with his username and password when he wants it. Once registered, the user logs in with their registered data.

As soon as the user has logged in, you will find the main window, where by default a panel is visible with the four different categories of events offered here. You can navigate any category. Entering one of them, the user can access any event, seeing more detailed information. In addition to seeing all the events, the user has the possibility to mark them as "favorite". This option allows the user to save events in another section, in order to know which ones have attracted more attention and to be able to access them in a quicker and direct way at the moment that it wants. To bookmark, the user has to touch the ASPA icon in the bottom right corner of each event.

Within each event the user has the following additional information:

- Accessibility services: They are indicated in text format accompanied by an icon to make their identification easier.
- Location: It contains the address of the place where the event is held, the phone and a direct access to its website.

ACA has accessible events in almost all cities in Spain, so the user has the option to filter events by city. It is possible that in the city that the user searches there are no events, but here it will communicate it, **Giving you an option to try your search again in another city (not yet implemented)**. To expedite this search, ACA has a section of "popular cities", which are the busiest. You can still enter the city name if it is not in this section.

Whenever you want to filter by city, you also need to indicate a type of accessibility: subtitling, audiodescription, sign language, or full accessibility. You will not be able to apply these filters separately, or apply them in different cities or different services accessible at the same time.

A useful and accessible method for the user is to be able to return to the main window from any point of the application, without having to go through the historical windows that he has visited. Touching the logo of the application, which is centered on the top of almost all the windows, it turns to the main one. The windows that do not have the logo, is because they have another way also simple to return to the beginning of the application.

The events that the user has saved as favorites are located in *My file*. This tab, which is in the main window, lists the events that the user stored, separated by genre. From this tab you can also access the additional information of each event. You can also unmark it as a favorite from this same site as from the event pane of each category (you may uncheck it as long as you have previously marked it). If you unmark it, it will disappear from this section, and you have the option to save it again whenever you want. **(Unfinished Deployment)**.

For the user to be aware of who are the organizers of this initiative, there is in the main window a section called *Cesya*. This shows information that makes this organization known, accompanied by a graph with the relationship of *Cesya* Between different organisms. (not finished)

The user, in any window, can access the side menu. This menu facilitates browsing through the application, because it has shortcuts to:

- *My profile*— A window that contains the user's personal information. From here you can modify this information, taking it to the window *Configuration*. The user can log off from here.
- *My file*: window where the events you have marked as favorites are
- *About the*: Window *Cesya*, to learn about the creative organization of ACA
- *Contact*: Direct communication with *Cesya*, either by email or social networks
- *Logout*: Exit your session and return to the initial application window

The window *Configuration* It contains another form where the user will be able to change his personal data, modifying the information of his account.

A registered user You can enter the application as a guest user, but it is imperative to have closed your session.

To explain in a more graphic way the scope of the registered user system, a diagram has been designed:

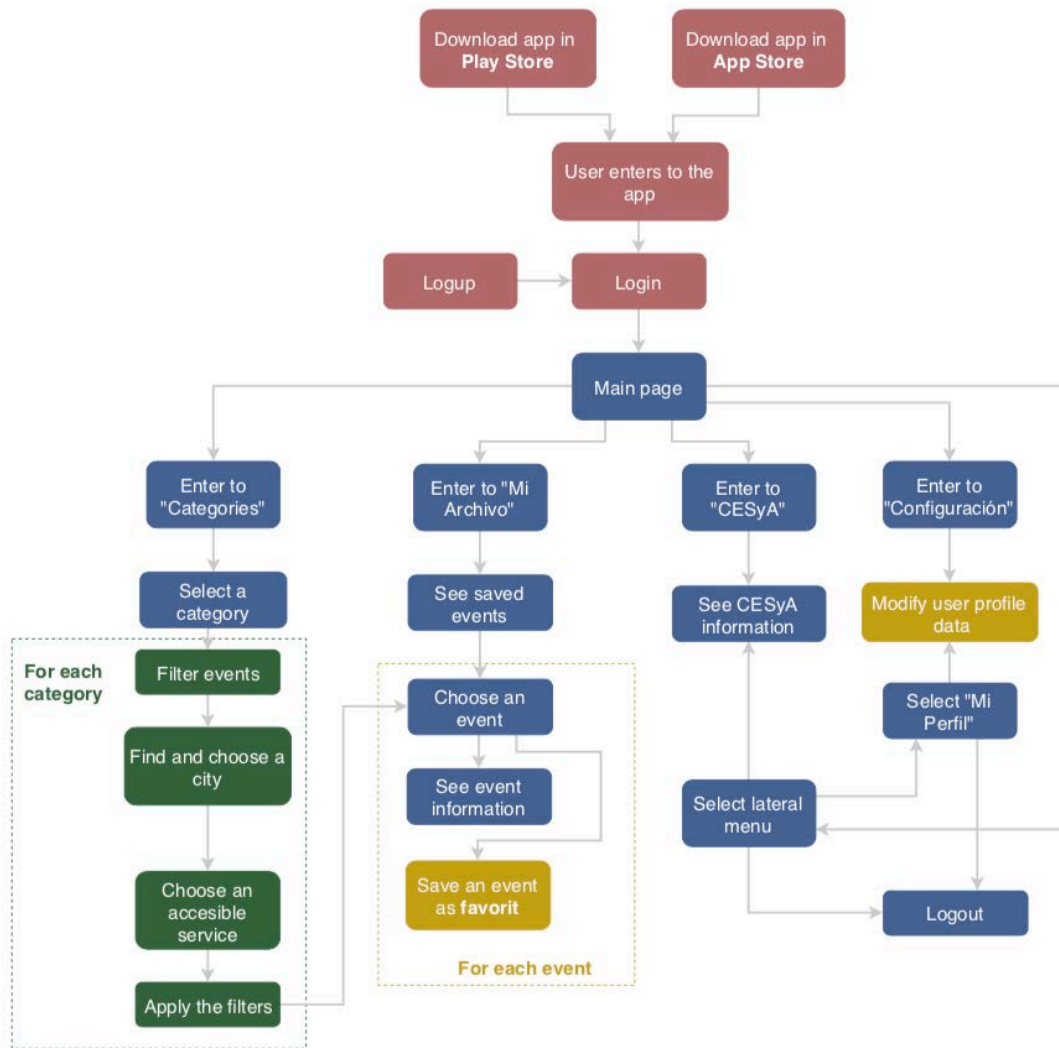


Illustration 72: Scope of the registered user system

2.1.2.2 Scope of System: Guest user

A guest user is the one who prefers not to create an account to access the application. This type of user implies having restricted certain services offered here. However, you have enough services in guest mode to be able to enjoy the application.

When a guest user enters the application, he or she can also access the "Log in" section, but because he does not have an account, he will not be able to continue. Therefore, the only viable way to enter is "to enter without registering."

The guest user has access to the panel of the four event categories. You can access them, seeing all the events of each one of them. The events shown by default are those available in Madrid with complete accessibility, without the possibility of applying any filter.

If the user wants to mark an event as a favorite, the ASPA icon is visible in each event. If it touches the icon, here you will be advised to register. The same thing happens with the tab *My file*, in the main window. You have access to it, but you can only see a message indicating that it is logged in order to store events.

A useful and accessible method for the user is to be able to return to the main window from any point of the application, without having to go through the historical windows that he has visited. Touching the logo of the application, which is centered on the top of almost all the windows, it turns to the main one. The windows that do not have the logo, is because they have another way also simple to return to the beginning of the application.

For the user to be aware of who are the organizers of this initiative, there is in the main window a section called *Cesya*. This shows information that makes known to this organization, accompanied by a graphic with the relation of CESyA between different organisms.

The user, in any window, can access the side menu. This menu facilitates browsing through the application, because it has shortcuts to:

- *Register*: The user will have the option to access the registration form from this menu
- *My file*: window where the events are, which, if logged, has bookmarked
- *About the*: Window *Cesya*, to learn about the creative organization of ACA
- *Contact*: Direct communication with CESyA, either by email or social networks

To explain in a more graphic way the scope of the guest user system, a diagram has been designed:

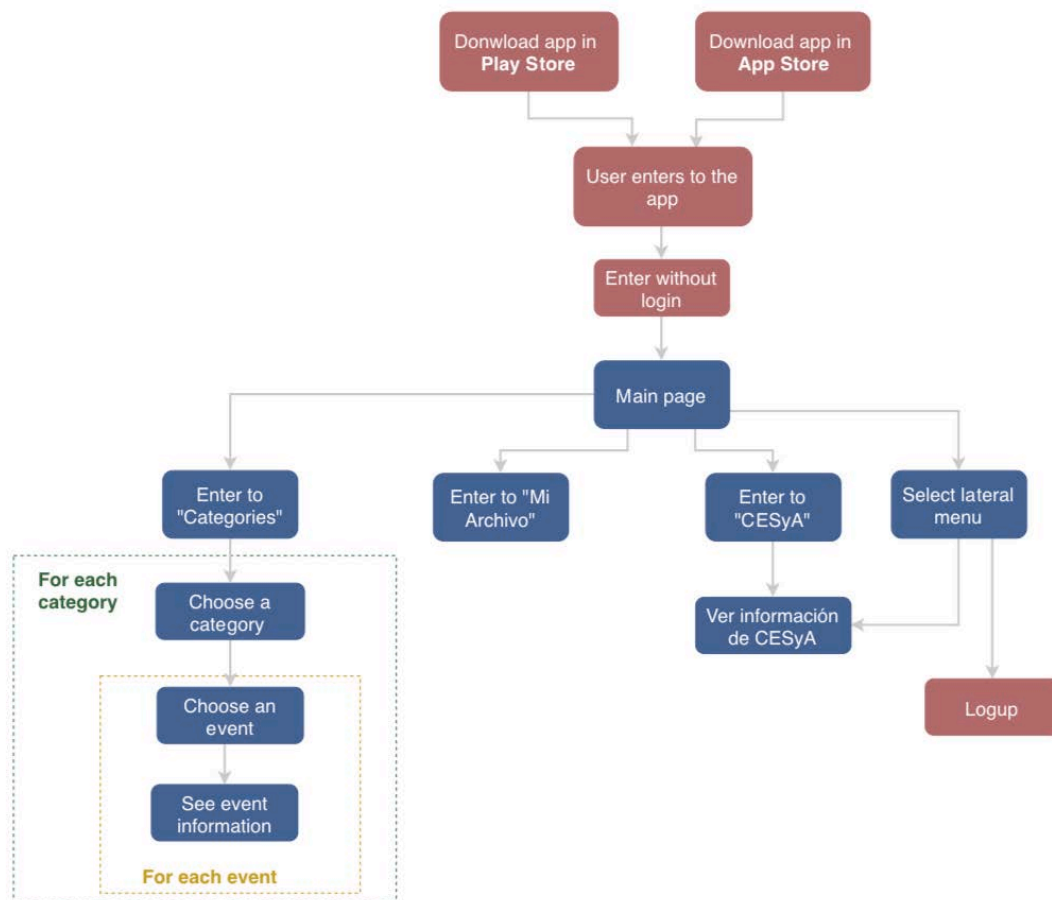


Illustration 73: User System Scope Guest

2.2 Study of the current situation

This section consists of the study of the current environment in which the system will be developed. In this way you will be able to identify in an optimal way the problem existing in the current systems, to find solution to it.

For this assessment are going to analyze existing applications that may be competition, studying why they are competing and the differences that exist between them. Depending on this assessment will take into account the advantages and disadvantages that the application has here, being able to enhance or confront them.

2.2.1 Assessing the current situation

To create the initial context of this study will expose the main problem by which the application will be developed. Similar applications that have this problem are also going to be detailed to understand it in a better way.

2.2.1.1 Competing applications

Accessible Cultural Agenda-- Accessible

This application is the original source of this project. It is aimed at people with hearing and visual impairment, who want to enjoy events with accessible services. It has four different types of events: Museum, theatre, cinema and others. The application has a simple operation, since it only offers the information of the services, filtered by city and type of accessibility.

This system is developed in a native way, that is, it has a development for each one the different operating systems in which it is deployed (*Android E ios*).

TOUR4all – PREDIF

TOUR4all is an application for the organization of accessible events. Provides reliable information on events in different cities. It is destined to use it when the users do tourism. Share all kinds of events, whether from finding adapted hotels to wheelchair-accessible transportation included. It also has information about cultural events.

Without knowing for sure if it is a multiplatform or native application, since it is not known the development of this application, it can be considered competition of ACA, by encompassing the same information. However, TOUR4all covers much more information since the end of the application is not the same.

2.2.1.2 problem of the current situation

Currently, there is no application of accessible events that your development serves for multiple platforms. The application *Accessible* It does not have a cross-platform development, since it is deployed separately in *Android* And in *ios*. It also presents a design with scarce Functionality. On

the other hand, the TOUR4all application also does not cover the whole objective that you want to cover with this project, because it is oriented towards a different purpose.

The current problem, therefore, is the high cost of maintaining both applications for the different types of operating systems.

It is a problem in terms of economic cost, because in this way we are making two native applications totally independent (even if it is the same), and development, because it is doubling the effort and work. In addition, when updating, upgrading or modifying the system, these changes must be taken into account in both developments. This can create discrepancies between the two applications in the face of the user, and it should not be given that the administration of the application must be transparent to the end user.

Another problem not so striking, is the little functionality of applications that offer information similar to the one you want to develop. As to *Accessible*, the only thing it provides is event information. Brings what is necessary, but it could be in a way Most optimal and complete. It also has a low usability because it does not follow many design canons. For example, it does not use an attractive design to make it pleasant to use for the end user, nor does it show that the user has thought about the design of the interface, making it easy and more usable any service offered by the application. An application of accessible events must be in the hands of usability.

AlsoWho finds the problem of scalability. A native application, developed for different platforms, presents low scalability by having to treat several systems for the same development. It can lead to loss of quality and it is difficult to grow in a fluid way, because it is not a homogeneous system.

3. System Analysis

This section aims to obtain a detailed specification of the system to be built, serving as a basis for the subsequent design.

It introduces the formal definition of the user's needs, as well as the specifications of the standards and standards that the application must follow. In addition, this task will study the subsystems in which the entire system is divided, achieving cohesion between them. It will also define the user interfaces that will define the application, Trazándolas with the actions that the user can perform on it.

Finally, this section will study the data model that follows the system.

3.1 System Definition

The ACA application is defined as a cultural agenda that provides accessible events for people with hearing and visual disabilities. This system provides the user with events that have accessible events in a given city with a certain accessibility service.

The problem that this project is going to cover is the cost of maintaining a native application. That's why What The solution to this problem will be to create a new multiplatform application covering of functionality and establishing a design that meets the canons of user interfaces.

3.2 Identification of analysis subsystems

This section defines the subsystems that compose the project system. Four subsystems can be distinguished:

- Access to the application
- Navigating the Application
- Event Filter
- Data storage

These subsystems follow the MVC pattern (model, view and controller), being:

- **Model:** Data storage Subsystem
- **View:** Application Navigation Subsystem
- **Driver:** Application access and event filter subsystems

The structure of the system, following the MVC pattern, would be as follows:

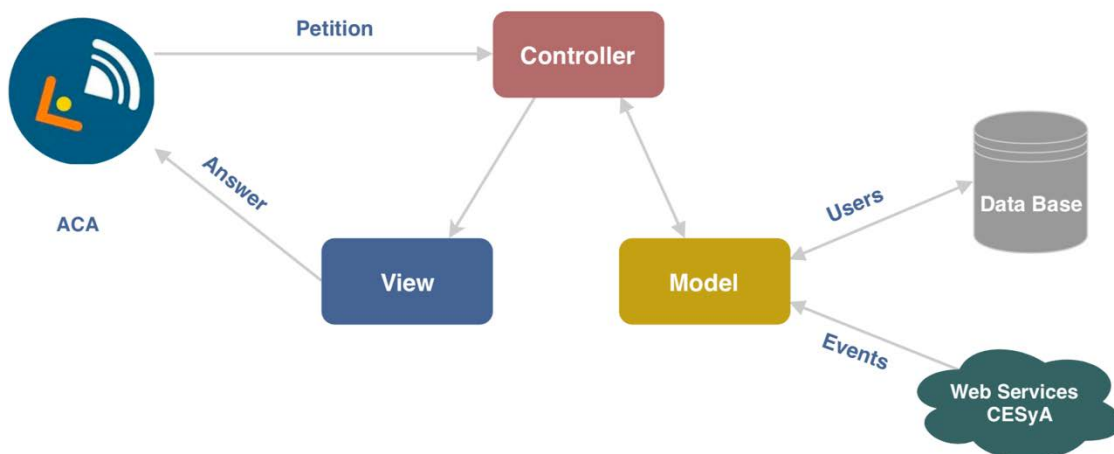


Illustration 74: System MVC Pattern

3.2.1 Definition of each subsystem

3.2.1.1 access to the application

This subsystem manages the accesses to the application. As defined above, the user can access the system having been registered or in guest mode.

This component will be responsible for acting as a *Driver* To manage user access to the system.

3.2.1.2 Navigation by Application

This subsystem treats the user's browsing by the application. Has the *View* Of the system as it is in charge of showing the user the answers to their requests.

3.2.1.3 Event Filter

This subsystem manages the filter that the user applies to events. is part of the *Driver* Of the system, because it has to manage the requests linked to the data that are obtained from the *Web Services*.

3.3.1.4 Data storage

This subsystem manages the *Model* of the system. It includes both the storage of user information in the database, and the reading of the data and the *Web Services*, that contains the event information. This extraction and loading of the information is specified in more detail in the definition of the data model.

Provides persistence of system data.

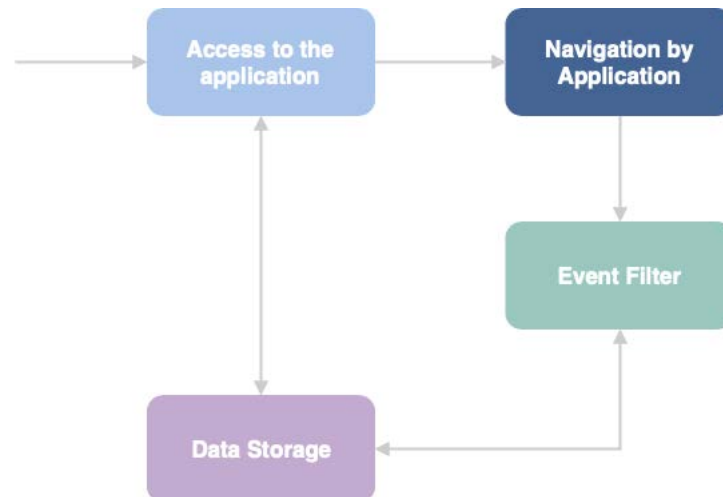


Illustration 75: Relationship between the subsystems

4. System Design

4.1 Objective

This section will solve the problem that has been described and modeled in the system analysis. The construction specifications will be generated relative to the same.

To specify this design, it will detail both the system architecture and its behavior to fully understand the operation and development of the application.

4.2 System architecture

This system presents an MVC architecture, which is the origin structure of the chosen alternative for its development, Apache Córdoba.

The project will be developed in a prototype-oriented language, which involves a different treatment to the object-oriented language. This implies that the architecture of the system is not structured in classes, y Therefore, do not have a class model or object instances.

Well it is true that from the specifications of **EcmaScript 6** It has included the management of classes in JavaScript, but this project has decided to follow this structure because the orientation to prototypes is much more flexible. However, the relationship that exists between each element of the architecture that follows the system is going to be designed.

Each prototype will be responsible for managing the *Driver* Of the system, and contain properties and variables to nourish logic to the rest of the modules of the architecture.

Both the *View* As the *Model* They are managed by different components of the system.

5. Conclusions

Once this project is over, it is a pleasure to comment on the conclusions that have been drawn.

To begin with, it is a very positive point that students have to face this type of work individually. It brings a greater perspective of reality, away from the academic that can be a work of the university. It provides a major and necessary responsibility for the working world, because you learn to face real problems, and to solve, or at least, to try.

On the other hand, this project had some barrier of entry, for example the ignorance of the development environment in which the whole system has been defined. It takes several months to secure this tool. Well it is true that it is not a complicated environment, but it is totally different from everything seen in the grade, so it was starting from scratch. However, this has involved a new knowledge and therefore has expanded the range of options to choose when you want to define some other system.

Another entry barrier was the total ignorance of the application creation for mobile devices. It is a very interesting area but it had never been touched. It was known web development, but not for mobile.

The chosen development environment has demanded that you do not leave the programming of these three languages in particular: HTML, JavaScript and CSS. The student had advanced knowledge about these programming languages, since they have been seen in various subjects. However, although it has been helpful to have a pre-base, we have learned new features and ways of developing, especially in JavaScript. For example Never before Requests were addressed to Web services with this language. This has helped reinforce knowledge about how a Web service works, how it reads and extracts information from them, etc. This is another thing that has brought new knowledge to the student.

For all this project have been able to apply many knowledge that have been learned in the race. In general, what has been most touched has been learned in subjects related to software engineering, such as *Principles Of Development Software PDS*, *So Development Project ManagementfTware (DPDS)* And *Software Engineering IS*, among others. Above all, it is appreciated to have studied subjects like DPDS because it shows the complete realization of the development of a software PRODUCT. Thanks to what has been learned in it, this project has been able to be done in a more comfortable way, because it has been the development of a software project from beginning to end, as it was seen in the subject.

It has also been possible to implement everything learned in subjects such as *Computer Technologies for the WEB Or Interfaces of UUser*. On the other hand, although during the degree consider of the most complex, all those subjects in which the administration and the design of data bases are treated Has been very useful for this project, for having provided a more concrete image of the data management. Thanks to them, the use of databases in this project not only has not been a problem, but it has been very pleasant to know how to manage them.

Thanks to the realization of this project, it has been possible to see the importance of the analysis and the design of an information system. In any software development there may be the idea of implementing the product without having to analyze how to do it or how to design it before implementing it. At least in the case of the student, it was a very abstract thing to understand. Having carried out this project, it has been verified in the first person the need to analyze and design a whole system before building it.

It is very gratifying to have learned to develop mobile applications, y In addition, to have learned a new paradigm of definition of mobile applications, as is the multiplatform development. This has made the developer here see his taste for development both Web and mobile. This is a fundamental point, since it helps the student's professional development, for having seen in what branch of everything he has learned he wants to perform his career.

It is hoped that this application, in the future, can help, in a real area, all people who need some kind of accessibility.

5.1 Possible system upgrades

This section will describe the possible improvements that could be applied to the system to improve its performance, usability and functionality.

- Allow the user to unsubscribe from the ACA systems: Right now the application does not have the possibility to unsubscribe the user in case you want. This function would bring more freedom to the user.
- Event Rating: Add star-format punctuation to events to serve as a *Feedback* Both for the users and for the creators of the event, being able to apply improvements.

- Apply to the development of the system the regulations and directives of an application accessible: currently ACA offers accessible services, but does not follow an accessible development. Create a development that meets all the terms of accessibility.
- Add audio to the application: include voice notes on the application that serve as a guide in browsing, for those who cannot see the application.
- Improve the database structure: Add new features and collections to the database.
- Improve the usability of the application: Continue adding features that make ACA a 100% usable application for all types of users.